



**RECIBE**  
Revista electrónica  
DE COMPUTACION, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA



# Índice

Editorial

Presentación

## Computación e Informática

- Gestión de Materiales Experimentales y su Uso en Replicaciones de Experimentos: Un Mapeo Sistemático de la Literatura  
(Replication Management Tools and Materials for Use in Replication in Experimental Software Engineering: A Systematic Mapping Study) I  
Edison Espinosa  
Juan Marcelo Ferreira
- Seguimiento de proyectos de programación. Una aplicación de GitHub en la educación  
(Programming projects monitoring. Using Github on education) II  
Javier Salazar Zárate  
Blanca Hidalgo Ponce  
Narcisa Salazar Alvarez  
Byron Vaca Barahona
- Sistema de navegación reactiva difusa para giros suaves de plataformas móviles empleando el Kinect  
(Fuzzy mobile reactive navigation system for smooth turns by using kinect) III  
Claudia Cruz Martínez  
Ignacio Algreto Badillo  
J. Jesús Arellano Pimentel  
Ernesto Cortés Pérez  
Francisco Aguilar Acevedo  
Luis Alberto Morales Rosales
- Los Sistemas Gestores de Flujos de Trabajo en la Gestión de Procesos Software IV  
(The Workflow Management Systems for Software Process Management)

Víctor Hugo Menéndez Domínguez  
María Enriqueta Castellanos Bolaños

Evidencia Empírica de la Minería de Procesos en la  
Implantación de CMMI-DEV  
(Empiric Evidence of Process Mining in CMMI-DEV  
Implementation)

Paola E. Velazquez-Solis  
Brenda L. Flores-Rios  
María Angélica Astorga-Vargas  
Jorge E. Ibarra Esquer  
Félix Fernando González Navarro  
Francisco J. Pino

V

Asistente escolar para los estudiantes de Ingeniería  
de Software: una aplicación móvil  
(School Assistant for Software Engineering students:  
a mobile application)

Julio César Díaz Mendoza  
Juan Pablo Ucán Pech  
Antonio Aguilera Gúémez  
Angélica Beatriz Toscano de la Torre

VI

## Biomédica

Potencial eléctrico en el corazón: representación  
mediante un grafo  
(Heart electrical potential: a graph representation)

Natalia González-Cervantes  
Aurora Espinoza-Valdez  
Ricardo Salido-Ruiz

VII

# Presentación

En este número tuvimos el gusto de tener como editores invitados a **Omar S. Gómez y Raúl H. Rosero**, de la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo (ESPOCH), de Riobamba, Ecuador, miembros del comité organizador del Simposio Iberoamericano en Programación Informática (SIIPRIN'2016) realizado en ESPOCH del 30 de junio al 1 de julio del 2016.

El SIIPRIN'2016 tuvo como intención el fomentar el contacto, la cooperación científica y profesional en la programación informática. En ésta, su primer edición, se recibieron un total de 23 trabajos con tasa de aceptación del 52%. De entre los trabajos aceptados, nuestros editores invitados seleccionaron dos para presentar una versión extendida en este número de la revista: “Gestión de Materiales Experimentales y su Uso en Replicaciones de Experimentos: Un Mapeo Sistemático de la Literatura”, de los autores Edison Espinosa y Juan Marcelo Ferreira, que como el título lo indica, presenta un estado del arte; y “Seguimiento de proyectos de programación. Una aplicación de GitHub en la educación”, de Javier Salazar Zárate y colegas, que presenta un método para la ejecución de dicha tecnología.

Otros editores invitados en este volumen: **Raúl Aguilar Vera**, de la Universidad Autónoma de Yucatán, y **Reyes Juárez Ramírez** de la Universidad Autónoma de Baja California, seleccionaron dos trabajos de la Conferencia Internacional de Ingeniería del Software, Investigación e Innovación, CONISOFT 2016, que se llevó a cabo en la ciudad de Puebla, México del 27 al 29 de abril del presente año. Los trabajos seleccionados y que se publican en este número, no aparecen en las memorias de dicho congreso. El primero: “Evidencia empírica de la minería de procesos en la implantación del CMMI-DEV” de Paola E. Velazquez-Solis y colegas es un caso práctico; y el segundo “Asistente escolar para los estudiantes de Ingeniería de Software: una aplicación móvil” de Julio César Díaz Mendoza y colegas, es un desarrollo tecnológico.

En este volumen se publican además dos artículos más de Computación e Informática y uno de Biomédica.

Esperamos que nuestros lectores los encuentren interesantes.

Adriana Peña Pérez Negrón  
Editor General.

# **Gestión de Materiales Experimentales y su Uso en Replicaciones de Experimentos: Un Mapeo Sistemático de la Literatura**

## **Replication Management Tools and Materials for Use in Replication in Experimental Software Engineering: A Systematic Mapping Study**

Edison Espinosa

Universidad de las Fuerzas Armadas , Ecuador

egespinosa1@espe.edu.ec

Juan Marcelo Ferreira

Facultad Politécnica de la Universidad Nacional de Asunción, Paraguay

jmferrera1978@gmail.com

**Resumen:** La Ingeniería de Software Experimental (ISE) aplica la experimentación para obtener conocimiento sobre productos, métodos, técnicas usados en el proceso de desarrollo software. Para validar el conocimiento es necesario repetir el experimento original tantas veces como sea necesario. Se requieren mucha información y materiales acerca del experimento para que un experimentador realice una replicación. Previa a la ejecución de la replicación del experimento, algunos o la totalidad de los materiales pueden requerir cambios produciendo nuevas o modificadas versiones de estos. Una vez ejecutada la replicación del experimento se espera

que, parte o la totalidad de los materiales, se incorporen al material del experimento original. El incremento en el número de replicaciones del experimento original está directamente relacionado con el incremento de las versiones del material experimental, que ocasiona comúnmente confusión y desorden en su organización. En este trabajo, realizamos un mapeo sistemático de la literatura (MSL) para localizar trabajos sobre Gestión de Materiales Experimentales y su Uso en Replicaciones en Ingeniería de Software Experimental. Los resultados encontrados reflejan la escasa existencia de artículos sobre esta temática. Además, el análisis de los estudios encontrados ha descubierto que dichos artículos, en su mayoría, adolecen de problemas en la gestión de versiones de materiales experimentales para poder llevar a cabo un proceso de replicación. Estos datos proporcionan información de interés para iniciar la investigación de la adopción del paradigma de gestión de configuración de software en la gestión del material experimental en ISE.

**Palabras clave:** Ingeniería del Software Experimental, Replicación de Experimentos, Material Experimental, Gestión del Material Experimental

**Abstract:** Experimental Software Engineering (ESE) applies experimentation to get knowledge about products, methods and techniques used in the software development process. The original experiment should be repeated as many times as necessary to validate that knowledge. Lots of information and materials about experiment is required for an experimenter to perform a replication. Before the execution of the experimental replication, all or part of the materials may require changes producing new or modified versions of these. After experimental replication is executed, it is expected that all or part of the materials are incorporated in the material of the original experiment. The increase of the number of replications of the original experiment is directly related to the increase of the versions of the experimental material, generating commonly confusion and disorder in the organization. In this work, we conduct a mapping study to locate articles about Replication Management Tools and Materials for use in experiment replication in ESE. The results show the lack of

articles in this subject. Furthermore, most of them suffers from problems in the version management of the experimental materials to carry out a replication process. These data provide interesting information to start a research of the adoption of the software configuration management paradigm in the experimental material management.

**Keywords:** Experimental Software Engineering; Experiment Replication; Experimental Material; Experimental Material Management.

## 1. Introducción

En la Ingeniería del Software (IS) se han realizado centenares de experimentos que abarcan virtualmente todas las áreas (Sjøberg et al., 2005) y este número no cesa de incrementarse (Dieste, Fernandez, Martinez, & Juristo, 2011; Zendler, 2001). La experimentación en IS es compleja, pero necesaria y requiere mecanismos para sustentar y motivar los estudios e integrar resultados; lo que a su vez, necesita de una comunidad de experimentadores que se centre en replicar estudios (V. R. Basili, Shull, & Lanubile, 1999). La replicación se utiliza para validar los resultados empíricos publicados por investigadores en IS (Daly, Brooks, Miller, Roper, & Wood, 1994; F. J. Shull, Carver, Vegas, & Juristo, 2008). Se requieren mucha información y materiales acerca del experimento para que un experimentador realice una replicación (Schmidt, 2009). Previa a la ejecución de la replicación del experimento, algunos o la totalidad de los materiales pueden requerir cambios produciendo nuevas o modificadas versiones de estos. Una vez ejecutada la replicación del experimento se espera que parte o la totalidad de los materiales se incorporen al material del experimento original. El incremento en el número de replications del experimento original está directamente relacionado con el incremento de versiones del material experimental, ocasionando comúnmente confusión y desorden en su organización donde el problema de gestionar las versiones de los materiales experimentales y las replications de un experimento se torna más compleja (Mendonça et al., 2008). La gestión de los materiales experimentales es una

cuestión abierta en ISE y de importancia notoria según han manifestado diversos autores (Do, Elbaum, & Rothermel, 2005; Gallardo, 2012; F. Shull et al., 2002; F. Shull, Mendonça et al., 2004). La literatura relacionada sobre la replicación de experimentos permite identificar la evolución de los instrumentos que se emplearon en la transmisión de información para replicar experimentos como: reportes de los experimentos, paquetes de replicación e infraestructura tecnológicas para apoyo a la experimentación (Vegas, Juristo, Moreno, Solari, & Letelier, 2006). En este artículo se presenta un MSL que se ha llevado a cabo para localizar investigaciones en los que se evidencie la gestión del material experimental y su uso en replicación de experimentos en ISE.

El artículo está estructurado de la siguiente manera. En la sección 2 se describen los antecedentes. En la sección 3 se describen la pregunta de investigación y la metodología. En la sección 4 se detalla el proceso de planificación y ejecución de las iteraciones. Los resultados son detallados en la sección 5. En la sección 6 se discuten aspectos relevantes de los artículos localizados y seleccionados. Las situaciones que amenazan la validez de esta investigación son presentadas en la sección 7. Finalmente, en las secciones 8 y 9 se dan las conclusiones y líneas futuras de la investigación.

## 2. Antecedentes

Para incentivar los procesos de experimentación y replicación en IS, algunos investigadores propusieron y desarrollaron instrumentos para transferir información, como publicaciones, paquetes de replicación y repositorios de experimentos (Gallardo, 2012). En el inicio, las publicaciones del experimento resultaban en un documento que se transfería para replicar el experimento (Kamsties & Lott, 1995). Posteriormente, investigadores como Lott (Lott,s.f.), Basili y colegas (V. Basili et al., 1996) entre otros propusieron y desarrollaron paquetes estáticos de replicación que contienen algunos documentos (materiales, objetos, instrumentos, formularios, instrumentos, especificaciones entre otros) que son necesarios para replicar un experimento. Finalmente,

algunos trabajos de investigación proporcionan infraestructuras tecnológicas para almacenamiento de materiales, instrumentos, documentos entre otros en diferentes medios, que son usados para transferir información. Estos pueden ir desde un repositorio de material experimental hasta la definición e implementación de herramientas para soportar las actividades del proceso experimental. Además, ha habido intentos para almacenar resultados (grupo de experimentos, casos de estudio y experimentos individuales) en lo que podría llamarse como bases de experiencia CEBASE (Boehm & Basili, April 2001), Visek (Hofmann & Wulf, 2002), Giants (Vanschoren & Blockeel, 2009) and SIR (Petersen, Feldt, Mujtaba, & Mattsson, 2008).

Los materiales experimentales en ISE podrían ser considerados como un conjunto común de productos: interrelacionados, reutilizables, almacenados en repositorios y que se usan para realizar futuros procesos de experimentación y replicación. Estos productos evolucionan en el tiempo debido: al cambio de contexto, por defectos encontrados en el material, la evolución de la investigación entre otros, que genera nuevas o modificadas versiones del material experimental. Los materiales experimentales y sus diferentes versiones se constituyen en el medio de comunicación entre los diferentes experimentadores. En este contexto proponemos llevar a cabo un MSL que permita localizar trabajos en los que se evidencie la gestión de materiales experimentales y su uso en replications de experimentos (Espinosa, 2014).

### 3. Pregunta de Investigación y Metodología

Dirigimos el alcance de esta investigación sobre artículos relacionados a la gestión de materiales experimentales y su uso en replications hasta julio de 2014. La pregunta de investigación que planteamos es:

**RQ1:** ¿Qué investigaciones existen sobre gestión de materiales experimentales para la realización de replications?

La RQ1 que nos planteamos en esta investigación busca localizar los artículos donde se identifique indicios de la gestión de versiones en el material experimental. Realizamos un MSL (Systematic Mapping Study, SMS en inglés) con dos iteraciones. Para cada iteración se desarrollaron las actividades de planificación y ejecución para identificar la cantidad, tipo de publicaciones y los reportes técnicos disponibles sobre esta temática. Según Petersen (Petersen et al., 2008), los pasos para llevar a cabo un mapeo sistemático son: la definición de la pregunta de investigación, la búsqueda de artículos relacionados, el análisis de resultados, la selección de artículos, el keywording del resumen y la extracción y mapeo de los datos. En cada uno de estos pasos se producen ciertos resultados y al final del proceso se tiene el mapa de los artículos relacionados sobre la temática. Además, la ejecución de estos pasos requiere el uso de recursos humanos, materiales, técnicos etc. Los recursos utilizados para la búsqueda de artículos incluyen un conjunto de reglas, actividades, elementos e instrumentos, que formalizan y dan soporte a las actividades de:

1. Selección de fuentes de artículos
2. Definición de la cadena de búsqueda
3. Definición de los criterios de inclusión y exclusión
4. Preselección de artículos
5. Selección de artículos
6. Extracción de datos de los artículos

Para la **selección de fuentes de artículos** establecemos un conjunto de reglas para garantizar la calidad y cantidad de los artículos. Las reglas que consideramos son: definición de criterios de selección de fuentes y motores de búsqueda. En la actividad de **definición de cadenas de búsqueda** se estructuraron las cadenas de búsqueda utilizadas para responder a la pregunta de investigación. Como lo propone Beecham (Beecham, Baddoo, Hall, Robinson, & Sharp, 2006), la cadena de búsqueda se construye a partir de la pregunta de investigación planteada con la estrategia PICOC y para refinar la pregunta de investigación se aplica la estrategia propuesta por Petticrew & (Petticrew & Roberts, 2008). Para la **definición de los criterios de inclusión y exclusión** se establecen previamente un conjunto de criterios utilizados durante la actividad de ejecución de la búsqueda para la preselección y selección de

artículos. En la actividad de **preselección de artículos** se procede a la lectura de algunas secciones del artículo a fin de identificar en el texto términos utilizados en la cadena de búsqueda. En la actividad de **selección de artículos** se hace una lectura completa del artículo, donde se busca evidenciar la aplicación de los términos de la cadena de búsqueda en el artículo. Para la **extracción de datos de los artículos** se crea un instrumento para recolectar los datos de los artículos relacionados con la temática. .

## 4. Planificación y Ejecución de las Iteraciones

Hemos realizado el mapeo sistemático de la literatura para la localización de artículos en dos iteraciones. En cada iteración se realizan tareas de planificación y ejecución que se describen a continuación.

### 4.1 Iteración I

En la planificación se formaliza el soporte a las actividades de organizar, ejecutar y entregar resultados de la búsqueda de artículos y en la ejecución se realizan las actividades de buscar, preseleccionar y seleccionar los artículos sobre la temática.

**Planificación** En esta primera iteración para la selección de fuentes de artículos se estableció como reglas: 1) los artículos deben proceder de fuentes fiables como conferencias internacionales, revistas, capítulos de libros, artículos de revistas científicas disponibles en la web y, 2) los motores de búsqueda son Scopus e IEEE. En la actividad de definición de las cadenas de búsqueda se estructuran las cadenas de búsqueda utilizadas para responder a la pregunta de investigación. Siguiendo la propuesta de Beecham (Beecham et al., 2006), la cadena de búsqueda se construye a partir de la pregunta de investigación planteada con la estrategia PICOC: Population, Intervention, Comparison,

Output, Context. Para responder a la pregunta de investigación planteada la estrategia PICOC tiene la composición que se muestra en la Tabla 1.

La definición de la estructura de la cadena se inicia identificando los términos para la población, intervención y el contexto. Usamos el término Experimentation para la población junto con los sinónimos de la cadena propuesto por Dieste (Dieste & Padua, 2007). Para la intervención se incluyen diversos términos utilizados para referirse al material en la ejecución de experimentos como: material en (Juristo & Vegas, 2011; Maldonado et al., 2006; F. Shull, Lanubile, & Basili, 2000), objects en (Do et al., 2005; Sabaliauskaite, Matsukawa, Kusumoto, & Inoue, 2003), artifacts en (Albayrak & Carver, 2012; V. R. Basili et al., 1999; He & Carver, 2006; F. Shull et al., 2004), instruments en (da Silva et al., 2013), components en (Dunsmore, Roper, & Wood, 2002) y laboratory package en (F. Shull et al., 2004). Para finalizar, en el contexto incluimos el término sharing considerando que el proceso experimental es un marco de trabajo en el que se comparten información, conocimiento y materiales entre experimentadores que buscan minimizar costos y maximizar la producción de conocimiento científico. En la Tabla 1 se muestra la lista de sinónimos relacionados con los términos de la pregunta de investigación unidos con el operador OR.

**Tabla 1.** Sinónimos y términos relacionados de búsqueda de artículos PICOC Iteración I.

PICOC	Término	Sinónimo/Término relacionado
<b>Población</b>	Experimentation	Experiment OR Empirical OR Empirical study OR Empirical evaluation, Experimentation OR Experimental comparison OR Experimental analysis OR Experimental evidence, Experimental setting OR Empirical data
<b>Intervención</b>	Experimental Material	Material OR Objects OR Artifact, Instruments OR Components OR Laboratory package

**Comparación** No aplicable

**Salida** No aplicable

**Contexto** Management Management OR Sharing

Finalmente, la población, intervención y contexto se unen con el operador AND. La cadena de búsqueda resultante es:

SRCHManagement= (Experiment OR Empirical OR Empirical study OR Empirical evaluation OR Experimentation OR Experimental comparison OR Experimental analysis OR Experimental evidence OR Experimental setting OR Empirical data) AND (Material OR Objects OR Artifact OR Instruments OR Components OR Laboratory package) AND (Management OR Sharing).

Para la definición de los criterios de inclusión y exclusión de artículos se utiliza la pregunta de investigación planteada. Estos criterios son utilizados durante la actividad de ejecución de la búsqueda para la preselección y selección de artículos. En la actividad de preselección de artículos, los criterios de inclusión permiten incorporar artículos en los que se evidencie el uso de términos de la cadena de búsqueda en las secciones: título del trabajo, resumen y palabras claves. Por lo contrario los criterios de exclusión permiten descartar los artículos en los que se utilizan los términos de la cadena de búsqueda pero que no corresponden al contexto de experimentación. Esta actividad genera una lista que contiene los estudios primarios de la búsqueda. Durante la actividad de selección de artículos se toman los criterios de inclusión para incorporar los artículos en los que se evidencie actividades de gestión de configuración sobre al menos un material experimental. Encambio, se aplican los criterios de exclusión para descartar los artículos en los que no se detallan actividades de gestión sobre al menos un material experimental, para ello se realiza una lectura completa del artículo. Esta actividad genera una lista que contiene los estudios seleccionados.

En la actividad de extracción de datos se identifican informaciones básicas de los artículos seleccionados. Para soportar esta actividad creamos el instrumento para registrar la información sobre: autores, el título del artículo, la fuente y el año de publicación y también los materiales experimentales sobre los que se realizaron las tareas de gestión. La Tabla 2 muestra el instrumento desarrollado.

**Tabla 2.** Instrumento para la extracción de datos de artículos seleccionados.

**Autores**

**Título del Artículo**

**Fuente de Publicación**

**Año de Publicación**

**Materiales Experimentales Identificados**

**Ejecución** En esta sección se describe la realización de las actividades previamente planificadas para localizar artículos sobre la temática. La ejecución comienza con el acceso a las bases digitales de Scopus e IEEE, seguidamente se selecciona la opción búsqueda avanzada en cada uno de los motores de búsqueda y a continuación se ingresa la cadena especificada en la definición de la cadena de búsqueda. Con la cadena de búsqueda SCRHManagement se ha localizado un total de 40 artículos en Scopus y 31 en IEEE. Luego de aplicar los criterios de inclusión y exclusión especificados para la preselección de artículos, han quedado 3 artículos de Scopus y 1 de IEEE. Estos resultados se visualizan en la Tabla 3.

En términos de la precisión de la búsqueda, determinado por la relación entre el número de artículos preseleccionados y localizados, el 7,5 % corresponde a Scopus, mientras que 3,22 % para IEEE.

Finalmente, luego del proceso de selección, tres artículos fueron excluidos de los cuatro preseleccionados por no cumplir alguno de los criterios establecidos.

**Tabla 3.** Artículos localizados, preseleccionados y precisión de la cadena SCRHManagement.

Cadena de búsqueda	Motor	Artículos Localizados	Artículos Preseleccionados	Precisión
SCRHManagement	Scopus	40	3 [STW+05], [SMB+04], [FPCM98+05]	7,5%
	IEEE	31	1 [VZJ03]	3,22%
<b>Total</b>		71	4	

En la Tabla 4 se detalla la justificación de la exclusión. En general los artículos excluidos no evidencian en su contenido la aplicación de actividades de gestión sobre materiales experimentales.

**Tabla 4.** Detalle de artículos incluidos y excluidos en la Iteración I.

#	Artículos Preseleccionados	Excluido	Justificación
1	[STW+ 05]	SI	El estudio aborda la problemática, de la gestión de materiales de enseñanza.
2	[STW+ 04]	NO	
3		SI	El trabajo implementa un prototipo sobre prácticas de gestión de requisitos.
4		SI	La investigación aborda la problemática de la clasificación, almacenamiento y recuperación de componentes de negocio que se utilizan en el ensamblaje de aplicaciones software.

Durante la actividad de extracción de datos se utiliza el instrumento definido en la planificación. La Tabla 5 muestra los datos del artículo [SMB+ 04] finalmente seleccionado

**Tabla 5.** Datos del Estudio Seleccionado en la Iteración I.

<b>Autores</b>	Shull F., Mendonca M.G., Basili V., Carver J., Maldonado J.C., Fabbri S., Travassos G.H., Ferreira M.C
<b>Título del Artículo</b>	Título del Artículo
<b>Fuente de Publicación</b>	Journal Of Information Science and Engineering
<b>Año de Publicación</b>	2004
<b>Materiales Experimentales Identificados</b>	Paquete de Laboratorio (Artefacto)

## 4.2 Iteración II

Como en la Iteración I, siguiendo el proceso riguroso hemos obtenido resultados muy limitados, apenas un estudio seleccionado, consideramos necesaria una segunda iteración donde replanteamos y realizamos ajustes en la planificación y ejecución. Por un lado, necesitamos aumentar la precisión de la cadena de búsqueda de la iteración I. Recuerde que la precisión es la relación entre los artículos preseleccionados y el total de artículos localizados en los motores de búsqueda. Por otro lado, debemos mejorar la cantidad de artículos encontrados sobre la temática. Para mejorar la precisión y la cantidad de artículos en esta iteración, hemos introducido variantes en la planificación, que se detallan a continuación.

**Planificación** En esta actividad consideramos importante mantener invariantes algunas reglas, elementos y actividades de la Iteración I. En este sentido, se va a mantener la pregunta de investigación, los criterios de inclusión y exclusión para las actividades de preselección y selección de artículos. Por otro lado hemos extendido los motores de búsqueda agregando a Scopus e IEEE los motores ACM y SpringerLink. Además, en esta iteración hemos incorporado un

segundo contexto, con el fin de incrementar la cantidad de artículos. Denominamos Contexto1 (management) a la referida en la Iteración I y Contexto2 (Support) a la Iteración II. Los valores de la estrategia PICOC para esta nueva iteración tiene la composición que se muestra en la Tabla 6.

La incorporación del Contexto2 ha producido dos cambios en la cadena de búsqueda. El primer cambio realizado es la modificación parcial de los términos (MPT), en la que únicamente se mantienen cuatro sinónimos de población, tres de intervención y adicionalmente se incorporan los sinónimos de soporte del Contexto2. El segundo cambio consiste en la modificación total de términos (MTT) donde en el campo población se usa únicamente el término Experimentation, en el campo intervención utilizamos la expresión Software Engineering y para el contexto se emplea un conjunto de expresiones sobre Support para el proceso de replicación experimental. En la Tabla 6 se detallan los sinónimos y términos separados con el operador OR para la MPT y MTT. Como veremos en la ejecución, la estrategia MTT surge como consecuencia de que los resultados obtenidos con la MPT no mejoran la precisión ni la cantidad de artículos localizados.

**Tabla 6.** MPT y MTT para la búsqueda de artículos PICOC.

	<b>PICOC</b>	<b>Término</b>	<b>Sinónimo / Término Relacionado /Expresión</b>
<b>MPT</b>	Población	Experimentation	Experimentation OR Experiment OR Experimental OR Empirical
	Intervención	Experimental Material	Material OR Objects OR Artifact
	Comparación	No aplicable	
	Salida	No aplicable	
	Contexto 1	Management	Management OR Sharing

	Contexto 2	Support	(Replication Support )2.1 (Package OR Lab Package OR Laboratory Package OR Replication Package )2.2 (Replication Infrastructure )2.3 Replication Repository )2.4
<b>MTT</b>	Población	Experimentation	Experimentation
	Intervención	Software engineering	Software engineering
	Comparación	No aplicable	
	Salida	No aplicable	
	Contexto 1	Support	(Experimental Material OR Experimental Material Replication OR Experimental Artifacts OR Experimental Object OR Experiment Material )1.1 (Experimental Package OR Lab Package OR Laboratory Package OR Replication Package)1.2 (Infrastructure OR Replication Infrastructure OR Experimental Infrastructure )1.3 (Repository OR Replication Repository OR Experimental Repository OR Experiment Repository )1.4 (Replication Support )1.5 (Replication Management OR Experimental Management OR Experimental Material Management OR Experimental Material Sharing)1.6

Finalmente para MPT se unen con el operador AND los términos y sinónimos resultando cinco cadenas de búsqueda, construidas de la siguiente manera:

1. SRCHManagement = Población AND Intervención AND Contexto1
2. SRCHSupportMan(i) = Población AND Intervención AND Contexto2.i, donde  $i=\{1..4\}$

Para MTT se unen con el operador AND los términos y sinónimos. Las cadenas de búsqueda MTT construidas son seis y se obtienen mediante la siguiente composición:

SRCHSupport(i)= Poblacion AND Intervención AND Contexto1,i, donde  $i=\{1..6\}$

**Ejecución** La actividad de búsqueda de estudios primarios comienza con el uso de la cadena de búsqueda con MPT en las bases digitales de Scopus, IEEE, ACM y SpringerLink. Luego en cada uno de los motores de búsqueda se selecciona la opción búsqueda avanzada y a continuación se ingresa las cadenas de búsqueda: SCRHManagement y SRCHSupportMan(i) con  $i=\{1..4\}$ . La cadena de búsqueda SCRHManagement ha localizado un total de 33 artículos en Scopus, 44 en IEEE, 299 en ACM y 21 en SpringerLink. Los resultados de las demás cadenas de búsqueda se ven en la Tabla 7.

**Tabla 7.** Artículos localizados, preseleccionados y precisión en Scopus, IEEE, ACM y SpringerLink con MPT.

Motor	Cadena de Búsqueda	Artículos Localizados	Artículos Preseleccionados	Precisión
<b>Scopus</b>	SCRHManagement	33	3 [BZS+ 07],[TMN+ 99], [BSL99]	9,0 %
	SCRHSupportMan1	4	1 [BSL99]	25,00 %
	SCRHSupportMan2	6	1 [TMN+ 99]	16,66 %
	SCRHSupportMan3	6	0	0,00 %
	SCRHSupportMan4	8	0	0,00 %
	Total		57	
<b>IEEE</b>	SCRHManagement	44	0	0,00 %
	SCRHSupportMan1	2	0	0,00 %

	SCRHSupportMan2	11	0	0,00 %
	SCRHSupportMan3	8	0	0,00 %
	SCRHSupportMan4	8	0	0,00 %
	Total	73	0	0,00 %
<b>ACM</b>	SCRHManagement	299	0	0,00 %
	SCRHSupportMan1	10	0	0,00 %
	SCRHSupportMan2	16	1 [MCDdO06]	6,25 %
	SCRHSupportMan3	26	0	0,00 %
	SCRHSupportMan4	42		
	Total	393	1 [ABF04]	2,38 %
<b>SpringerLink</b>	SCRHManagement	21	0	0,00 %
	SCRHSupportMan1	33	1 [BSL00]	3,03 %
	SCRHSupportMan2	24	0	0,00 %
	SCRHSupportMan3	13	0	0,00 %
	SCRHSupportMan4	14	1 [BSL00]	7,14 %
	Total	105		

Con los artículos localizados en la búsqueda se aplican los criterios de inclusión/exclusión definidos en la actividad de preselección de artículos explorando el uso de los términos de la cadena de búsqueda en las secciones: título, resumen y palabras claves. En la Tabla 7 se muestran los resultados de la actividad de preselección. El total de artículos localizados en Scopus fue de 57, ACM 393, IEEE 73 y Springer 105. La precisión de la búsqueda de los artículos en la biblioteca digital: Scopus se encuentra en el rango del 0-25 %, IEEE se encuentra en 0 %, ACM se encuentra en el rango de 0-6,25 % mientras

que SpringerLink tiene un rango de 0-7,14 %. Para la selección, se lee el artículo completo buscando evidencias de actividades de gestión en los materiales experimentales en el estudio analizado.

Recuerde que necesitamos mejorar los resultados de la búsqueda y bajo esta premisa la MPT no proporciona progreso alguno. De ahí, decidimos cambiar completamente los términos de búsqueda y surge la estrategia MTT obligando a replantear la cadena de búsqueda en la planificación. Una vez definidos los nuevos términos de búsqueda se procede a la ejecución. Para realizar la búsqueda de estudios primarios utilizando la nueva cadena de búsqueda con MTT accedemos a las bases digitales de Scopus, IEEE, ACM y SpringerLink, seguidamente en cada uno de los motores de búsqueda seleccionamos la opción búsqueda avanzada y a continuación ingresamos las cadenas de búsquedas SCRHSupport(i) con  $i=[1..6]$ . En la actividad de preselección de artículos se realiza la revisión de los artículos para buscar el uso de los términos de la cadena de búsqueda en las secciones: título, resumen y palabras claves. En la Tabla 8 se muestran los resultados de la actividad de preselección. El total de artículos localizados en Scopus fue de 120, IEEE 260, ACM 881, y Springer 182.

**Tabla 8.** Artículos localizados, preseleccionados y precisión en Scopus, IEEE, ACM y SpringerLink con MTT.

Motor	Cadena de Búsqueda	Artículos Localizados	Artículos Preseleccionados	Precisión
<b>Scopus</b>	SCRHSupport1	10	0	0,00 %
	SCRHSupport2	11	3 [MCDdO06], [SCVJ08], [SMB+04]	27,27 %
	SCRHSupport3	19	1 [SCVJ08]	5,26 %
	SCRHSupport4	8	0	0,00 %

	SCRHSupport5	72	1 [BSL99]	1,38 %
	SCRHSupport6	0	0	0,00 %
<b>Total</b>		120		
<b>IEEE</b>	SCRHSupport1	119	4 [MMdO+ 08], [MCDdO06], [BSL99], [SBC+ 02]	0,33 %
	SCRHSupport2	2	1 [SBC+02]	50,00 %
	SCRHSupport3	20	2 [LT09], [TdSNB08]	10,00 %
	SCRHSupport4	33	2 [LT09], [JV09]	6,06 %
	SCRHSupport5	10	2 [BSL99], [MA03]	20,00 %
	SCRHSupport6	76	2 [LT09], [BCL+ 01]	2,63 %
<b>Total</b>		260		
<b>ACM</b>	SCRHSupport1	263	0	0,00 %
	SCRHSupport2	4	2 [SBC+ 02], [JV09]	50,00 %
	SCRHSupport3	179	1 [ABF04]	0,55 %
	SCRHSupport4	209	0	0,00 %
	SCRHSupport5	7	0	0,00 %
	SCRHSupport6	219	1 [JV09]	0,45 %
<b>Total</b>		881		
<b>SpringerLink</b>	SCRHSupport1	115	4 [SMB+ 04], [DER05], [BZS+ 07], [BSL00]	0,34 %

SCRHSupport2	0	0	0,00 %
SCRHSupport3	0	0	0,00 %
SCRHSupport4	67	2 [SMB+ 04], [BSL00]	2,98 %
SCRHSupport5	0	0	0,00 %
SCRHSupport6	0	0	0,00 %
<b>Total</b>	<b>182</b>		

La precisión de la búsqueda de los artículos en la biblioteca digital: Scopus se encuentra en el rango del 0-27,27 %, IEEE se encuentra en el rango de 0,33-50 %, ACM se encuentra en el rango de 0-50 % mientras que SpringerLink tiene un rango de 0-2,98 %.

En la selección de artículos se realiza una revisión de los artículos para evidenciar el uso de la gestión en al menos un material experimental. En la Tabla 9 se muestra los artículos preseleccionados. Nótese, que el total de artículos excluidos en esta iteración ha sido 12 de los 16 preseleccionados por no cumplir alguno de los criterios establecidos. En general los artículos excluidos no evidencian en su contenido la aplicación de actividades de gestión sobre materiales experimentales.

**Tabla 9.** Detalle de artículos incluidos y excluidos en la Iteración II.

#	Artículos Preseleccionados	Excluido	Justificación
1	[TdSNB08]	SI	El estudio trata sobre la problemática del soporte al proceso de experimentación en ingeniería del software (ISE).

2	[BCL+ 01]	SI	El trabajo aborda la problemática de mantener una apropiada administración del conocimiento
3	[MMdO+08]	NO	
4	[ABF04]	SI	El trabajo aborda el problema sobre el desarrollo de software de alta calidad con tiempos y costos efectivos.
5	[BSL99]	SI	La problemática que aborda el artículo es la construcción de conocimiento en base a una familia de experimentos relacionados.
6	[TMN+ 99]	SI	El estudio aborda la problemática sobre el soporte al desarrollo del software.
7	[SMB+ 04]	NO	
8	[LT09]	SI	La problemática que aborda el estudio es sobre el manejo del gran volumen de información y el tiempo empleado en la ejecución de experimentos en ISE.
9	[BZS+ 07]	SI	El estudio aborda la problemática para obtener, usar, compartir y comparar datos recolectados de una variedad de fuentes.
10	[SBC+ 02]	No	
11	[DER05]	No	
12	[SCVJ08]	SI	El estudio aborda la problemática sobre el rol de la replications exactas en la construcción de conocimiento en la comunidad de ingeniería de software.
13	[BSL00]	SI	La problemática que aborda el artículo es la construcción de conocimiento en

base a una familia de experimentos relacionados.

14	[MCDdO06]	SI	La problemática que aborda el estudio es la dificultad de definir y correr experimentos controlados en ingeniería del software.
15	[JV09]	SI	La problemática que aborda el trabajo es sobre la obtención de conocimiento a través del uso de los resultados obtenidos de la ejecución de repeticiones no idénticas.
16	[MA03]	SI	El trabajo aborda la problemática para la creación de un modelo exacto de simulación.

La actividad de extracción de datos utiliza el instrumento creado en la actividad de planificación. Así se registra la información básica de los artículos en diferentes campos del instrumento, con excepción del artículo de Shull [SMB+04], que fue localizado en la iteración I. Los tres nuevos artículos seleccionados en la iteración II se visualizan en la Tabla 10.

**Tabla 10.** Datos de los artículos seleccionados en la iteración II.

1. Autores	Mendonca, Manoel G;Maldonado, José Carlos;de Oliveira, Maria Cristina Ferreira; Carver, Jeffrey; Fabbri, CPF;Shull, Forrest; Travassos, Guilherme Horta;Hohn, EN;Basili, Víctor R.
Título del Artículo	A framework for software engineering experimental replications
Fuente de Publicación	Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference
Año de Publicación	2008

Materiales Experimentales Identificados	Paquete de laboratorio, artefacto
<b>2. Autores</b>	Do, Hyunsook; Elbaum, Sebastian; Rothermel, Gregg
Título del Artículo	Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact.
Fuente de Publicación	Empirical Software Engineering
Año de Publicación	2005
Materiales Experimentales Identificados	Artefactos, objetos
<b>3. Autores</b>	Shull, F.; Basili, V.; Carver, J.; Maldonado, J.C.; Travassos, G.H. Mendonca, M.; Fabbri, S.
Título del Artículo	Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem.
Fuente de Publicación	ISESE '02: Proceedings of the 2002 International Symposium on Empirical Software Engineering.
Año de Publicación	2002
Materiales Experimentales Identificados	Paquete de Laboratorio (Artefactos)

## 5. Resultados

RQ1: ¿Qué investigaciones existen sobre gestión de materiales experimentales para la realización de replicaciones?

Como se muestra en la Tabla 10 se seleccionaron 4 artículos, donde se evidenció el detalle de actividades de gestión del material experimental. Específicamente, los estudios encontrados fueron: Mendonça [MMdO+ 08], Shull [SMB+ 04, SBC+ 02] y Do [DER05]. En [MMdO+ 08] y [SMB+ 04, SBC+ 02] los investigadores gestionan paquetes de laboratorio consistente en un contenedor de artefactos. En los estudios analizados, los artefactos son el material experimental. Para finalizar, en Do [DER05] se gestionan artefactos y objetos, siendo éste último el material experimental que se gestiona en la investigación.

## 6. Discusión

Desde el mapeo sistemático de la literatura que llevamos a cabo, concluimos que la situación sobre estudios donde se gestionan materiales experimentales para el uso en el proceso de replicaciones en ISE permitió: 1) localizar y analizar pocos artículos, a pesar de que el número de experimentos y replicaciones se han incrementado en las últimas décadas y, 2) eliminar otros trabajos por no cumplir con los criterios de exclusión definidos. Los artículos localizados y analizados fueron cuatro, que son investigaciones de fondo y consolidadas, pues existen varios reportes generados desde algunas décadas atrás y que inclusive se mantienen vigentes. El análisis de estos trabajos nos permitió identificar que todos adolecen de problemas en la gestión de versiones del material experimental, provocadas por el elevado número de replicaciones ejecutadas. En relación al número de trabajos excluidos es alta como se puede ver en las tablas 8 y 9 esto se debe a dos razones que identificamos. La primera es que en la preselección de artículos se buscó excluir aquellos donde se utilicen los términos de las cadenas de búsqueda pero que su uso no corresponda al contexto de experimentación, específicamente se buscó que gestionen materiales experimentales y no productos del desarrollo del software que fue común encontrar en los que los trabajos que se excluyeron. La segunda razón se relaciona con la actividad de selección que permitió excluir los artículos donde

no se evidencia indicios de actividades de gestión sobre algún material experimental. A continuación detallamos los artículos localizados.

En Shull [11] se aborda la problemática de la transmisión del conocimiento entre experimentadores para incrementar y mejorar las replicaciones experimentales, para lo cual proponen el uso de una estructura colaborativa basada en varios medios de comunicación y paquetes de laboratorio. Además Mendonca y colegas [8] describen la problemática sobre la transmisión de conocimiento entre grupos de investigación tanto internos como externos para llevar a cabo procesos de replicación de un experimento y proponen un marco de trabajo llamado FIRE (Framework for Improving the Replication of Experiments). En FIRE se direcciona el problema de compartir conocimiento entre investigadores fomentando la coordinación entre grupos de investigación con el uso de diferentes medios y mecanismos de comunicación, que a decir de los autores, facilitan la transmisión de conocimientos, permiten reducir costos, mejoran la calidad de las replicaciones y proveen resultados generalizables. También Shull y colegas [9] describen la problemática de la transferencia del conocimiento tácito entre experimentadores para realizar mejores procesos de replicación de experimentos, estudios complementarios y robustas conclusiones en un área específica de la ISE. La solución propuesta por los autores sustenta la idea de fomentar la relación entre el experimentador original y el replicador para transmitir el conocimiento tácito a través de una estructura colaborativa además del uso del paquete del laboratorio. Finalmente, Do y colegas [10] abordan la problemática de la baja cantidad de experimentos sobre la efectividad de las técnicas de testing, la escasa cantidad de datos empíricos y además sobre los costos que conlleva la ejecución de experimentos controlados en IS. La solución propuesta es una infraestructura para soportar experimentos controlados en técnicas de testing y testing de regresión. La infraestructura, descrita por los autores, permite la replicabilidad de experimentos, la agregación de resultados y la amortización de costos en experimentación.

El análisis de los artículos localizados permitió identificar indicios de aplicación de la gestión sobre materiales experimentales, pero en los estudios analizados los investigadores indican problemas en la gestión de versiones de materiales experimentales para poder llevar a cabo un proceso de replicación, específicamente hemos encontrado carencias con respecto a la información que permita conocer el estado y la traza de los materiales experimentales y la ausencia de actividades de gestión sobre los materiales experimentales.

## 7. Amenazas a la Validez

La principal amenaza está relacionada con las palabras claves utilizadas para la selección de los estudios. Los términos y sinónimos que empleamos para la construcción de las cadenas de búsqueda son comúnmente usadas en la comunidad que investiga en ISE. Esto lo realizamos mediante un estudio de base realizado sobre artículos sugeridos por expertos que conocen y tienen experiencia sobre experimentación y replicación en ISE. Una amenaza a la validez secundaria está relacionada con el factor humano al momento de aplicar los criterios de inclusión y exclusión para la preselección y selección de artículos. Para mitigar esta amenaza los investigadores realizaron un chequeo individual y, posteriormente, un chequeo cruzado. El chequeo individual consistió en que cada investigador revisa y genera una lista de artículos, posteriormente cada lista fue revisada para consolidarla en una única lista de artículos.

## 8. Conclusiones

Se ha realizado un estudio de mapeo sistemático con dos iteraciones en la que buscamos identificar la cantidad, tipo de publicaciones y reportes técnicos disponibles sobre la gestión del material experimental para su uso en replicaciones. En la primera iteración se ha obtenido un sólo artículo sobre la temática, motivando a realizar una segunda iteración donde se ha mantenido ciertos parámetros de la iteración I, entre ellas la pregunta de investigación, los criterios de inclusión y exclusión para las actividades de preselección y selección

de artículos. Además se ha añadido un segundo contexto con el objetivo de identificar estudios acerca del soporte al proceso de replicación de experimentos empleando gestión de materiales experimentales. Estos cambios, han conducido a la generación de dos nuevas cadenas de búsqueda. A la primera le denominamos modificación parcial de términos (MPT) en la que se ha mantenido algunos de los sinónimos empleados en la cadena original, mientras la segunda lleva la denominación de modificación total de términos (MTT) donde se han cambiado todos los sinónimos de los términos de la estrategia PICOC. Este nuevo escenario de MTP y MTT ha logrado encontrar cuatro artículos más incluido el artículo localizado en la iteración I. De ahí, se puede concluir la escasa existencia de artículos sobre la temática. Además, el análisis de los artículos encontrados ha descubierto que dichos artículos, en su mayoría, adolecen de problemas en la gestión de versiones de materiales experimentales para poder llevar a cabo un proceso de replicación. Finalmente, la gestión de materiales experimentales representa un marco novedoso y poco trabajado, y con este artículo hemos proporcionado información de interés como punto de partida para la investigación de la adopción del paradigma de gestión de configuración de software en la gestión del material experimental en ISE.

## 9. Trabajos Futuros

En el mapeo sistemático conducido, se realizó un esfuerzo sustancial para seleccionar un grupo de artículos a partir de una gran cantidad de estos. Creemos que futuras ampliaciones y actualizaciones se beneficiarán de nuestros esfuerzos, debido a que estas extensiones sólo tendrán que aplicar el procedimiento de búsqueda de las publicaciones después de 2014. Por lo que esperamos realizar actualizaciones continuas para este estudio de mapeo, usando o mejorando el protocolo de investigación que se describe en este artículo.

**Reconocimientos** Los autores agradecen a la Universidad de las Fuerzas Armadas (ESPE), al Grupo de Investigación en Ingeniería de Software Empírica

(GrISE), a la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) y a la Facultad Politécnica FP-UNA por el soporte brindado durante el desarrollo de este trabajo.

## Referencias

Albayrak, Ã., & Carver, J. (2012). Investigation of individual factors impacting the effectiveness of requirements inspections: A replicated experiment., 1-26. doi:10.1007/s10664-012-9221-0

Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sorumgard, S., & Zelkowitz, M. (1996). Packaging researcher experience to assist replication of experiments. Proc. of the ISERN Meeting 1996,

Basili, V. R., Shull, F., & Lanubile, F. (1999). Building knowledge through families of experiments.25(4), 456-473.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2006). Protocol for a systematic literature review of motivation in software engineering. University of Hertfordshire,

Boehm, B., & Basili, V. (April 2001). The CEBASE framework for strategic software development and evolution.

da Silva, F. Q. B., França, A. C. C., Suassuna, M., de Sousa Mariz, L. M. R., Rossiley, I., de Miranda, R. C. G., . . . Espindola, E. (2013). Team building criteria in software projects: A mix-method replicated study. Information and Software Technology, 55(7), 1316-1340. doi:http://dx.doi.org/10.1016/j.infsof.2012.11.006

Daly, J., Brooks, A., Miller, J., Roper, M., & Wood, M. (1994). Verification of results in software maintenance through external replication. Software Maintenance, 1994. Proceedings., International Conference On, 50-57. doi:10.1109/ICSM.1994.336790

Dieste, O., & Padua, O. A. G. (2007). Developing search strategies for detecting relevant experiments for systematic reviews. Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium On, 215-224. doi:10.1109/ESEM.2007.19

Dieste, O., Fernandez, E., Martinez, R. G., & Juristo, N. (2011). Comparative analysis of meta-analysis methods: When to use which? Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference On, 36-45. doi:10.1049/ic.2011.0005

Do, H., Elbaum, S., & Rothermel, G. (2005). Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Software Engineering*, 10(4), 405-435.

Dunsmore, A., Roper, M., & Wood, M. (2002). Further investigations into the development and evaluation of reading techniques for object-oriented code inspection. *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference On*, 47-57.

Espinosa, E. (2014). *Gestión de configuración y línea de productos para mejorar el proceso experimental en ingeniería del software*.

Gallardo, E. G. E. (2012). Using configuration management and product line software paradigms to support the experimentation process in software engineering. *Research Challenges in Information Science (RCIS), 2012 Sixth International Conference On*, 1-6. doi:10.1109/RCIS.2012.6240454

He, L., & Carver, J. (2006). PBR vs. checklist: A replication in the n-fold inspection context. *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, 95-104.

Hofmann, B., & Wulf, V. (2002). Building communities among software engineers: The VISEK approach to intra-and inter-organizational learning., 25-33.

Juristo, N., & Vegas, S. (2011). Design patterns in software maintenance: An experiment replication at UPM - experiences with the RESER'11 joint replication project. *Replication in Empirical Software Engineering Research (RESER), 2011 Second International Workshop On*, 7-14. doi:10.1109/RESER.2011.8

Kamsties, E., & Lott, C. M. (1995). An empirical evaluation of three defect-detection techniques. *Proceedings of the Fifth European Software Engineering Conference*,

Lott, C. M. Comparing reading and testing techniques. Retrieved from <http://web.archive.org/web/20071026204353/http://www.chris-lott.org/work/exp/>

Maldonado, J. C., Carver, J., Shull, F., Fabbri, S., Dória, E., Martimiano, L., . . . Basili, V. (2006). Perspective-based reading: A replicated experiment focused on individual reviewer effectiveness. *Empirical Software Engineering*, 11(1), 119-142.

Mendonça, M. G., Maldonado, J. C., de Oliveira, Maria Cristina Ferreira, Carver, J., Fabbri, C., Shull, F., . . . Basili, V. R. (2008). A framework for software engineering experimental replications. *Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference On*, 203-212.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. 12th International Conference on Evaluation and Assessment in Software Engineering, , 17 1.

Petticrew, M., & Roberts, H. (2008). Systematic reviews in the social sciences: A practical guide Wiley-Blackwell.

Sabaliauskaite, G., Matsukawa, F., Kusumoto, S., & Inoue, K. (2003). Further investigations of reading techniques for object-oriented design inspection. *Information and Software Technology*, 45(9), 571-585.

Schmidt, S. (2009). Shall we really do it again? the powerful concept of replication is neglected in the social sciences.13(2), 90-100.

Shull, F. J., Carver, J. C., Vegas, S., & Juristo, N. (2008). The role of replications in empirical software engineering. *Empirical Software Engineering*, 13(2), 211-218.

Shull, F., Basili, V., Carver, J., Maldonado, J. C., Travassos, G. H., Mendon\cca, M., & Fabbri, S. (2002). Replicating software engineering experiments: Addressing the tacit knowledge problem. ISESE '02: Proceedings of the 2002 International Symposium on Empirical Software Engineering, 7.

Shull, F., Lanubile, F., & Basili, V. R. (2000). Investigating reading techniques for object-oriented framework learning. *Software Engineering, IEEE Transactions On*, 26(11), 1101-1118. doi:10.1109/32.881720

Shull, F., Mendon\cca, M. G., Basili, V., Carver, J., Maldonado, J. C., Fabbri, S., . . . Ferreira, M. C. (2004). Knowledge-sharing issues in experimental software engineering.9, 111-137(27).

Shull, F., Mendonça, M. G., Basili, V., Carver, J., Maldonado, J. C., Fabbri, S., . . . Ferreira, M. C. (2004). Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9(1-2), 111-137.

Sjøberg, D. I., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N., & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *Software Engineering, IEEE Transactions On*, 31(9), 733-753.

Vanschoren, J., & Blockeel, H. (2009). Stand on the shoulders of giants. towards a portal for collaborative experimentation in data mining. Proceedings of the SoKD-09 International Workshop on Third Generation Data Mining at ECML PKDD 2009, , 1 88-99.

Vegas, S., Juristo, N., Moreno, A., Solari, M., & Letelier, P. (2006). Analysis of the influence of communication between researchers on experiment replication.

Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, 28-37.

Zendler, A. (2001). A preliminary software engineering theory as investigated by published experiments. *Empirical Software Engineering*, 6(2), 161-180.

## Apéndice: Lista de Artículos Preseleccionados

[ABF04] Roger T. Alexander, James M. Bieman, and Robert B. France. A software engineering research repository. *SIGSOFT Softw. Eng. Notes*, 29(5):1-4, September 2004.

[BCL+ 01] V. Basili, P. Costa, M. Lindvall, M. Mendonca, C. Seaman, R. Tesoriero, and M. Zelkowitz. An experience management system for a software engineering research organization. In *Software Engineering Workshop, 2001. Proceedings. 26th Annual NASA Goddard*, pages 29-35, 2001.

[BSL99] V.R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *Software Engineering, IEEE Transactions on*, 25(4):456-473, Jul 1999.

[BSL00] Victor R. Basili, Forrest Shull, and Filippo Lanubile. Using experiments to build a body of knowledge. In *Proceedings of the Third International Andrei Ershov Memorial Conference on Perspectives of System Informatics, PSI '99*, pages 265-282, London, UK, UK, 2000. Springer-Verlag.

[BZS+ 07] Victor R. Basili, Marvin V. Zelkowitz, Dag I. Sjøberg, Philip Johnson, and Anthony J. Cowling. Protocols in the use of empirical software engineering artifacts. *Empirical Softw. Engg.*, 12(1):107-119, February 2007.

[DER05] Hyunsook Do, Sebastian Elbaum, and Gregg Rothermel. Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Softw. Engg.*, 10(4):405-435, October 2005.

[FPCM98] P. Fowler, M. Patrick, A. Carleton, and B. Merrin. Transition packages: an experiment in expediting the introduction of requirements management. In *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on*, pages 138-145, 1998.

[JV09] Natalia Juristo and Sira Vegas. Using differences among replications of software engineering experiments to gain knowledge. In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM '09, pages 356–366, Washington, DC, USA, 2009. IEEE Computer Society.

[LT09] V.P. Lopes and G.H. Travassos. Knowledge repository structure of an experimental software engineering environment. In Software Engineering, 2009. SBES '09. XXIII Brazilian Symposium on, pages 32–42, Oct 2009. [MA03] J. Munch and O. Armbrust. Using empirical knowledge from replicated experiments for software process simulation: a practical example. In Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on, pages 18–27, Sept 2003.

[MCDdO06] Manoel Mendonca, Daniela Cruzes, Josemeire Dias, and Maria Cristina Ferreira de Oliveira. Using observational pilot studies to test and improve lab packages. In Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ISESE '06, pages 48–57, New York, NY, USA, 2006. ACM.

[MMdO+ 08] M.G. Mendonca, J.C. Maldonado, M.C.F. de Oliveira, J. Carver, S.C.P.F. Fabbri, F. Shull, Guilherme H. Travassos, E.N. Hohn, and V.R. Basili. A framework for software engineering experimental replications. In Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference on, pages 203–212, March 2008.

[SBC+ 02] Forrest Shull, Victor Basili, Jeffrey Carver, José C. Maldonado, Guilherme Horta Travassos, Manoel Mendonca, and Sandra Fabbri. Replicating software engineering experiments: Addressing the tacit knowledge problem. In Proceedings of the 2002 International Symposium on Empirical Software Engineering, ISESE '02, pages 7–, Washington, DC, USA, 2002. IEEE Computer Society.

[SCVJ08] Forrest J. Shull, Jeffrey C. Carver, Sira Vegas, and Natalia Juristo. The role of replications in empirical software engineering. *Empirical Softw. Engg.*, 13(2):211–218, April 2008.

[SMB+ 04] Forrest Shull, ManoelG. Mendonca, Victor Basili, Jeffrey Carver, JosC. Maldonado, Sandra Fabbri, GuilhermeHorta Travassos, and MariaCristina Ferreira. Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9(1-2):111–137, 2004.

[STW+ 05] Jun-Ming Su, Shian-Shyong Tseng, Ching-Yao Wang, Ying-Chieh Lai, Yu-Chang Sung, and Wen-Nung Tsai. A content management scheme in a

SCORM compliant learning object repository. J. Inf. Sci. Eng., 21(5):1053–1075, 2005.

[TdSNB08] G.H. Travassos, P.S.M. dos Santos, P.G.M. Neto, and J. Biolchini. An environment to support large scale experimentation in software engineering. In Engineering of Complex Computer Systems, 2008. ICECCS 2008. 13th IEEE International Conference on, pages 193–202, March 2008.

[TMN+ 99] K. Torii, K.-i. Matsumoto, K. Nakakoji, Y. Takada, S. Takada, and K. Shima. Ginger2: an environment for computer-aided empirical software engineering. Software Engineering, IEEE Transactions on, 25(4):474–492, Jul 1999.

[VZJ03] P. Vitharana, F.M. Zahedi, and H. Jain. Knowledge-based repository scheme for storing and retrieving business components: a theoretical design and an empirical analysis. Software Engineering, IEEE Transactions on, 29(7):649–664, July 2003.

## Notas biográficas:



**Edison G. Espinosa** es Ingeniero de Sistemas e Informática por la Universidad de las Fuerzas Armadas ESPE de Ecuador. Ha obtenido el Título de Doctor en Software y Sistemas en la Universidad Autónoma de Madrid, España. Se desempeña como docente investigador en la Carrera de Software del Departamento de Eléctrica y Electrónica. Sus áreas de interés son la Ingeniería de Software Experimental y la línea de producto software.



**Juan M. Ferreira** es Ingeniero en Informática y Licenciado en Análisis de Sistemas Informáticos en la Facultad Politécnica-UNA. Ha obtenido la Maestría en Software y Sistemas en la Facultad de Informática-Universidad Politécnica de Madrid. Es Especialista en Tecnologías de la Información con énfasis en Ingeniería del Software. Se desempeña como Director de Desarrollo en la Cámara de Senadores-Congreso Nacional. Sus áreas de interés son la Usabilidad e Ingeniería de Software Experimental.



Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 2.5 México.

## COMPUTACIÓN E INFORMÁTICA

*Recibido 4 Abr 2016 Aceptado 2 May 2016*

*ReCIBE, Año 5 No. 3, Noviembre 2016*

# Seguimiento de proyectos de programación. Una aplicación de GitHub en la educación

**Programming Projects Monitoring. Using Github on Education**

Javier Salazar Zárate<sup>1</sup>  
Javier.salazar@eduklip.com

Blanca Hidalgo Ponce<sup>2</sup>  
bhidalgo@epoch.edu.ec

Narcisa Salazar Alvarez<sup>2</sup>  
nsalazar@epoch.edu.ec

Byron Vaca Barahona<sup>2</sup>  
bvacab@esPOCH.edu.ec

<sup>1</sup>eduKlip, Ecuador.

<sup>2</sup>Escuela Superior Politécnica de Chimborazo, Ecuador.

**Resumen:** Esta investigación desarrolla y propone un método para la ejecución y seguimiento de los proyectos de fin de curso de las materias de lenguajes de programación en carreras de Ingeniería Electrónica soportado tecnológicamente por GitHub. El objetivo es obtener un método que permita mejorar la calidad de los proyectos. Para la validación del nuevo método se realiza un contraste entre dos grupos de estudiantes, el primer grupo utilizando el método tradicional de ejecución de proyectos y el segundo utilizando el método propuesto. Los resultados muestran que a pesar que la aplicación del nuevo método demanda de un esfuerzo adicional por parte del docente, evidentemente mejora el proceso de seguimiento de los proyectos de fin de curso: facilitando el trabajo colaborativo, permitiendo una evaluación objetiva, visibilizando los hábitos de estudio de los estudiantes, transparentando y automatizando las actividades inherentes a la ejecución de proyectos de software, potenciando el seguimiento y la guía a los estudiantes.

**Palabras clave:** Propuesta metodológica, Seguimiento de proyectos, Proyecto de fin de curso, Método [MESEPP], sistema de versionamiento [Github], lenguajes de programación.

**Abstract:** This research develops and proposes a method for the execution and monitoring of end course projects for programming languages topics imparted on Electronics Engineering Bachelor Degrees, this supported technologically by Github. The goal is to obtain a method to improve the quality of projects. To validate the new method, a contrast between two groups of students was done, for the first group the traditional method for project execution is used and for the second group, the new method is applied. The results show that even though the application of the new method demands an additional effort for the

teacher, it evidently improves the monitoring process of the end course projects by making collaborative work easy, allowing an objective evaluation, making visible student's habits, automating and making transparent the inherent activities of software projects execution, enhancing the student's monitoring and guidance.

**Keywords:** Methodological proposal, Project monitoring, End course project, method [MESSEP], Version control system [Github], Programming languages

# 1. Introducción

La utilización del proyecto de fin de curso en carreras de ingeniería es pertinente, sin embargo su aplicación trae consigo algunos problemas. El proyecto es un trabajo en equipo, sin embargo, algunos estudiantes siguen realizando un esfuerzo individual. El proyecto es un proceso, sin embargo, los proyectos se visibilizan al docente en un único día designado para la presentación del proyecto.

Entendiendo las particularidades de un proyecto de Lenguajes de Programación, en el que un programa de computadora, con líneas de código aportadas por cada uno de los miembros del equipo, es el producto resultante, el presente trabajo de investigación pretende definir, proponer y validar un método para el seguimiento del proyecto de fin de curso, con el fin de mejorar la calidad de los proyectos.

Como un apoyo tecnológico y a la vez como una parte integrante del método, se utiliza "GitHub", la plataforma para versionamiento de archivos más popular del mercado de software y de uso gratuito para proyectos de código abierto.

Con el desarrollo del trabajo se pretende dar respuesta a la siguiente pre-gunta de investigación:

¿Permitirá un método de seguimiento de proyectos de fin de curso aplicado en la materia de Lenguajes de Programación y soportado por GitHub, mejorar la calidad de los proyectos en carreras de Ingeniería Electrónica en la Escuela Superior Politécnica de Chimborazo?

A pesar de que la estimación de la calidad es subjetiva para diferentes escenarios, el presente trabajo sugiere una rúbrica para la ponderación de la calidad de un proyecto de fin de curso en las materias de Lenguajes de Programación.

En la sección 2 se conceptualiza a un método y se describe a GitHub, su importancia y los estudios relacionados en el ámbito de la educación. En la sección 3 se presenta y detalla el método propuesto, en la sección 4 se describe la metodología utilizada en el estudio para la aplicación y validación del nuevo método, en la sección 5 se discuten los resultados del estudio y finalmente en la sección 6 se presentan las conclusiones.

## 2. Antecedentes

Un método, como lo define la Real Academia de la Lengua es el “modo de decir o hacer con orden, modo de obrar o proceder, hábito o costumbre que cada uno tiene y observa” (Real Academia Española, 2001), puede ser definido como el “camino o procedimiento viable para conseguir un fin propuesto” (Gutiérrez, 2006; citado en Coria, et al. 2013) y difiere del concepto de metodología definido como “el estudio de los métodos” (Bunge, 2005). Un método refiere la forma en la que se puede realizar una actividad, tarea o trabajo para alcanzar un objetivo particular. Para ello un método puede pertenecer a alguno de los tipos existentes en las diferentes áreas de conocimiento (Sierra, 2001). El presente estudio propone entonces una forma de abordar la ejecución y seguimiento de los

proyectos de fin de curso en la materia de lenguajes de programación. Como parte integrante del método se incorpora la herramienta Github.

GitHub es una plataforma web comercial lanzada en el año 2008 (Yu, Yin, Wang, & Wang, 2014), diseñada para facilitar el almacenamiento centralizado, la trazabilidad y colaboración sobre archivos de proyectos. En su esencia GitHub registra los cambios que han ocurrido en el tiempo en uno o varios archivos, con la finalidad de recuperar, en el futuro, versiones específicas de ellos, esta es una característica propia de un sistema de control de versiones (Chacon S, 2014).

El corazón de GitHub es Git, una herramienta gratuita y de código abierto que se encuentra en la categoría de sistemas de control de versiones distribuido (Git, 2015). Actualmente GitHub mantiene más de 28.1 millones de repositorios (GitHub, Inc., 2015) y en la línea de software soporta proyectos de importancia mundial en una gran variedad de lenguajes de programación (Ray, Posnett, Filkov, & Devanbu, 2014).

Based on the previous statement, despite the vast quantity of tools to help urban modeling, there is a lack of tools devoted to simulate study-cases focusing in city operation, instead of its mere visualization; since modern tools frequently focus in 3D simulation and do not have enough power to deal with issues regarding how chaotic and erratic a real city is.

This first stage of this city simulator is aimed at the replication of open source projects. GitHub es gratuita para la ejecución de proyectos de código abierto y requiere de un pago periódico para proyectos de ámbito privado.

Hoy en día la plataforma es muy popular en las comunidades, especialmente de desarrolladores. Individuos y corporaciones utilizan GitHub como la herramienta primaria para el control y versionamiento de sus proyectos de software.

En este tipo de proyectos los miembros de un equipo se benefician de la plataforma logrando estrategias efectivas para coordinar los proyectos, mejorando sus habilidades técnicas, e inclusive gestionando su reputación (Dabbish, Stuart, Tsay, & Herbsleb, 2012).

En el ámbito de la educación existen algunas experiencias documentadas acerca del uso de GitHub. Y aunque esta plataforma no está diseñada especialmente para la educación, su "naturaleza social y colaborativa, es un recurso consistente con la ideología de educación liberal" (Shaffer, 2013).

Sus creadores conscientes del impacto que la plataforma puede tener en el área de la educación han puesto a disposición un portal educativo, en el cual se estima que son ya 1.200 cursos y 70.000 estudiantes registrados para beneficiarse del servicio (Sawers, 2014).

Si bien por su origen, el uso de GitHub en la educación parecería tener un campo de acción limitado a las carreras y materias dedicadas al desarrollo de software, algunos autores (Zagalsky, Feliciano, Storey, Zhao, & Wang, 2015), consideran que el uso de estas herramientas de versionamiento de código no debería estar restringido únicamente a estas áreas.

En el presente estudio se propone un método que incorpore a GitHub como el componente tecnológico que soporte el seguimiento de los proyectos de fin de curso. Las experiencias positivas del uso de GitHub y Git existentes en el ámbito de la educación (Xu, 2012), en la carrera de Ciencias de la Computación (Lawrance, Jung, & Wiseman, 2013) y específicamente en materias de lenguajes de programación (Josh, 2014), motivan y respaldan su elección para la nueva propuesta.

### 3. Propuesta

Entendido el objetivo de un método y habiendo respaldado el uso de la plataforma GitHub basado en experiencias positivas de su uso en el ámbito de

la educación, en la siguiente sección se presenta la propuesta de un método para el seguimiento de proyectos de fin de curso en materias de Lenguajes de Programación utilizando GitHub. Al nuevo método se lo ha denominado como MESEPP.

### 3.1 Procedimiento

El procedimiento sugiere el conjunto de pasos secuenciales sugeridos al docente para el proceso de seguimiento y evaluación de los proyectos de fin de curso. Los artefactos nombrados como parte del procedimiento serán detallados más adelante.

1. El docente de la materia de Lenguajes de Programación realiza la planificación del proyecto de fin de curso. Para ello completa el ARTEFACTO 01: Definición de proyecto.

2. El docente realiza un taller de capacitación de GitHub, de 3 horas de duración. En el taller se realizan las siguientes actividades:

- Presentación del ámbito de la herramienta, su objetivo, utilidad, proyectos importantes soportados por la plataforma. El objetivo es motivar al estudiante en el uso de la plataforma.
- Creación de cuentas individuales de los estudiantes en el portal de GitHub.
- Instalación de la herramienta GitHub de escritorio para el sistema operativo de preferencia del estudiante.
- Creación de un proyecto por equipo en la plataforma y vinculación de cada uno de los miembros.
- Explicación y práctica del uso de la herramienta GitHub para escritorio.
- Inclusión del docente como miembro de cada uno de los equipos.

3. Entrega del ARTEFACTO 02: Manual de GitHub a los estudiantes como complemento de la capacitación y como una fuente de consulta y referencia para los estudiantes.

4. El docente envía a un estudiante definido como el líder del equipo el ARTEFACTO 03: Definición de equipo, el mismo que es completado y enviado de regreso al docente.

5. El docente instala la aplicación GitHub para escritorio en el sistema operativo de su preferencia.

6. Desde el día acordado como la fecha de inicio del proyecto el docente diariamente, en lo posible a la misma hora del día, realiza las siguientes actividades:

- Descarga la última versión de cada uno de los proyectos de sus estudiantes.
- Revisa los cambios realizados de manera individual por cada miembro del equipo.
- Llena la información del ARTEFACTO 04: Retroalimentación.
- Una vez completada la revisión, envía por correo electrónico las retroalimentaciones a cada uno de los estudiantes. Los correos electrónicos serán los definidos en el ARTEFACTO 03: Definición de equipo. Es deseable realizar el envío por correo electrónico por ser un medio de comunicación personal. GitHub en su esquema gratuito expone de manera pública toda la información, mecanismo inconveniente para el envío de retroalimentaciones. Su exposición pública podría causar perjuicios por parte de otros miembros del mismo equipo, de otros equipos o de personas externas al proceso.
- El docente completa el ARTEFACTO 05: Registro de avance en las secciones individual y grupal para este día de revisión.
- En el caso que en el día de revisión exista un aporte grupal o individual de gran impacto o de interés técnico/pedagógico, el docente incluye esta información en un formato de pregunta en el ARTEFACTO 06: Cuestionario final.

7. El día definido como fecha de fin de ejecución del proyecto, el docente realiza la última revisión de los proyectos y envía las últimas retroalimentaciones.

8. En las fechas comprendidas entre la fecha de fin de ejecución del proyecto y la fecha de presentación final, el docente llena el ARTEFACTO 07: Rúbrica de evaluación en su sección “Durante la ejecución del proyecto”. El docente utiliza para ello la información recolectada en el ARTEFACTO 05: Registro de avance.

9. El docente valida que el ARTEFACTO 06: Cuestionario final se encuentre completo para cada uno de los equipos y estudiantes, en caso contrario lo completa en base al trabajo de seguimiento realizado sobre cada uno de los proyectos.

10. Se realiza la presentación final de los proyectos en la fecha definida para dicho evento. Se entrega a los estudiantes el tiempo definido para la presentación y defensa del proyecto.

11. Luego de la presentación, el docente realiza al menos una pregunta individual a cada uno de los miembros de los equipos. La pregunta es tomada del ARTEFACTO 06: Cuestionario final. De ser necesario el docente realiza las preguntas adicionales que crea necesarias.

12. El docente llena el ARTEFACTO 07: Rúbrica de evaluación en la sección “Presentación de proyecto”. Utiliza para ello la técnica de observación durante la presentación y la ronda de preguntas.

13. Una vez realizadas las presentaciones finales, el docente totaliza los valores del ARTEFACTO 07: Rúbrica de evaluación. El valor resultante es el grado de calidad del proyecto de cada equipo en su componente individual y grupal. Este valor que representa un porcentaje de 0% a 100% puede ser correspondido con la calificación final del proyecto. Así si un estudiante ha obtenido un valor del 70% para un proyecto cuya puntuación haya sido planificada para 10 puntos el puntaje correspondiente del estudiante será de 7.

### 3.2 Artefactos

En la siguiente sección se caracterizan cada uno de los artefactos mencionados como parte del procedimiento:

**Artefacto 01** - Definición de proyecto. Documento detallado del proyecto sugerido a todos los equipos de trabajo, incluyendo la siguiente información:

1. Título del proyecto.
2. Objetivos de la aplicación (software).
3. Objetivos en la materia.
4. Descripción general de la aplicación.
5. Requerimientos específicos.
6. Fecha de inicio del proyecto.
7. Fecha de fin del proyecto.
8. Presentación final, incluyendo fecha, hora, lugar y condiciones.
9. Método, describiendo el procedimiento detallado a los estudiantes, la forma de trabajo, las acciones que se esperan de ellos y la rúbrica de evaluación a utilizarse para la valoración del proyecto.

**Artefacto 02** - Manual de GitHub. Manual detallado de GitHub, incluyendo pantallas de la aplicación de escritorio, instrucciones para la creación de cuentas en el servicio web, creación del proyecto, estrategia de contribución y colaboración sobre el proyecto y resolución de conflictos.

**Artefacto 03** - Definición de equipo. Documento de definición de un equipo y sus miembros, incluyendo la siguiente información:

1. Equipo: Número de identificación única del equipo
2. Líder: Apellidos y nombres del miembro líder del equipo.
3. Integrantes: Detalle de cada uno de los integrantes del equipo, incluyendo: Código del estudiante, apellidos, nombres, nombre preferido, email y usuario GitHub.

**Artefacto 04** - Retroalimentación. Documento que concentre las retroalimentaciones diarias entregadas a cada uno de los miembros de los equipos, incluyendo la siguiente información:

1. Equipo: Número de identificación única del equipo.
2. Retroalimentaciones: De cada uno de los estudiantes, incluyendo: Apellidos y nombres, fecha y texto de la retroalimentación.

**Artefacto 05** - Registro de avance. Documento que permita registrar el avance grupal e individual de cada uno de los equipos, incluyendo la siguiente información:

1. Equipo: Número de identificación única del equipo.
2. Avance sobre el total del proyecto: Registro, en valor porcentual, del avance que el grupo realiza por día sobre el 100% esperado del proyecto.
3. Actividad individual: Registro por día de la actividad de cada miembro respecto al resto de los integrantes, utilizando las ponderaciones: alta, media, baja y ninguna.
4. Observaciones: Observaciones importantes a resaltar en el avance individual de los miembros del equipo.

**Artefacto 06** - Cuestionario final. Documento que recoge para cada miembro del equipo al menos una pregunta de importancia a ser preguntada y respondida el día de la presentación, incluyendo para cada estudiante:

1. Apellidos y nombres.
2. Antecedentes: Contextualizar la pregunta.
3. Pregunta.
4. Ámbito de respuesta: Enmarcar el ámbito de la posible respuesta esperada

**Artefacto 07** - Rúbrica de evaluación. Documento que permite ponderar el grado de calidad de los proyectos de fin de curso. La rúbrica tiene dos momentos de evaluación. El primero es durante la ejecución del proyecto y en él se pondera el 80% del total. Un segundo momento es el día de presentación final del proyecto, en este se pondera el 20% del total. En cada uno de estos momentos se tienen índices tanto a nivel grupal como a nivel individual.

Las siguientes tablas presentan el detalle de la rúbrica:

**Tabla 1.** Rúbrica de evaluación. Componente grupal durante la ejecución del proyecto.

	Valoración Grupal (60%)		
	0%	10%	20%
El equipo repartió entre sus integrantes equitativamente el esfuerzo del proyecto.	No. Algunos miembros del equipo hicieron más esfuerzo que otros.	En parte. En cantidad de líneas de código la repartición fue casi equitativa, pero algunos ejecutaron los requerimientos más complejos y otros sólo los más sencillos.	Sí. El equipo repartió el proyecto de manera equitativa tanto en cantidad como en complejidad del proyecto.
El equipo repartió equitativamente el tiempo dedicado al proyecto.	No. En los primeros días casi no se observó trabajo, mientras que en los últimos se completó la mayor cantidad del proyecto.	En parte. Se observó participación continua en el proyecto, pero no fue suficiente para completar todos los requerimientos.	Sí. Se observó un trabajo continuo y sostenido desde los primeros días hasta los últimos. El proyecto fue terminado. Es más se terminó el proyecto antes de la finalización de los días previstos para su ejecución.
El equipo ha completado todos los requerimientos especificados para el proyecto.	No. No se han completado todos los requerimientos.	Parcialmente. No se han completado todos los requerimientos, pero los ejecutados funcionan bien.	Sí. Se han completado todos los requerimientos y todos funcionan bien.

**Tabla 2.** Rúbrica de evaluación. Componente individual durante la ejecución del proyecto.

	Valoración Individual (20 %)		
	0%	10%	20%
Las partes de código creadas, respetan la teoría y consideraciones presentadas en clase.	No. Se descuidaron temas y consideraciones tratadas en clase. El código no es óptimo. Es más tiene errores que ocurren u ocurrirán bajo ciertas condiciones.	Parcialmente. En algunos temas se han tomado en cuenta las consideraciones tratadas en clase, en otras secciones se han descuidado.	Sí totalmente. Se ha tenido cuidado en utilizar los temas y consideraciones tratados en clase. En algunos casos se han utilizado otras estrategias producto de una investigación del estudiante.

**Tabla 3.** Rúbrica de evaluación. Componente grupal en la presentación del proyecto.

	Valoración Grupal (10%)		
	0%	5%	10%
La presentación final del proyecto fue satisfactoria.	No. No se observa preparación de la presentación. Fue desordenada y no se dio realce al proyecto.	Parcialmente. Se nota una preparación algo improvisada de la presentación.	Sí. Se nota una preparación adecuada de la presentación. Tiene responsables asignados, tiempos coherentes y una secuencia de presentación que da realce al proyecto.

**Tabla 4.** Rúbrica de evaluación. Componente individual en la presentación del proyecto.

	Valoración Individual (10%)		
	0%	5%	10%
El estudiante respondió correctamente la pregunta realizada por el profesor.	No. La pregunta no fue respondida, la respuesta no se entendió o se respondió algo diferente a lo preguntado.	Parcialmente. Se respondió la pregunta parcialmente. No se dijeron cosas importantes esperadas en la respuesta.	Sí. Se respondió la pregunta total y coherentemente.

## 4. Metodología

En la presente investigación se ha considerado como grupo de estudio a los estudiantes de la materia de Lenguajes de Programación de las carreras de Ingeniería Electrónica de la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo (ESPOCH). Cada semestre existen nuevos estudiantes matriculados, por lo que esta población se la puede entender como de número indefinido.

Con el procedimiento se pretende validar que la aplicación del nuevo método permite mejorar la calidad de los proyectos de fin de curso. La Tabla 5 resume las características de los grupos muestrales tomados para la investigación.

**Tabla 5.** Grupos muestrales seleccionados para el estudio.

	Periodo 2014	Periodo 2015
<b>Escuela</b>	Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes	Escuela de Ingeniería Electrónica en Control y Redes Industriales
<b>Periodo Lectivo</b>	marzo 2014 – agosto 2014	marzo 2015 – agosto 2015
<b>Semestre</b>	Segundo	Segundo
<b>Materia</b>	Lenguajes de Programación I	Lenguajes de Programación I
<b>Materias que los estudiantes reciben en Primer Semestre</b>	Matemática I Algebra lineal Física I Fundamentos de programación Química Metodología de la investigación	Matemática I Algebra lineal Física I Fundamentos de programación Química Metodología de la investigación
<b>Materias que los estudiantes reciben en el Segundo Semestre</b>	Matemática II Física II Lenguajes de programación I Herramientas EDA Sistemas operativos Sistemas digitales I	Matemática II Física II Lenguajes de programación I Herramientas EDA Sistemas operativos Sistemas digitales I
<b>Número de estudiantes</b>	28	28
<b>Método de seguimiento de proyectos de fin de curso</b>	Tradicional	Propuesto

El grupo del periodo marzo – agosto 2014 y el grupo del periodo marzo – agosto 2015 serán referidos en esta y en las siguientes secciones simplemente como Periodo 2014 y Periodo 2015 respectivamente.

Nótese que a pesar de que las Escuelas de Ingeniería Electrónica de las cuales se han tomado los grupos de estudio son diferentes, las condiciones de las muestras son iguales:

- El número de estudiantes es de 28 para los dos grupos.
- Los dos grupos reciben tanto en el primero como en el segundo semestre las mismas materias, los mismos contenidos y el mismo número de horas semanales en cada una de las materias.

- Antes del primer semestre los estudiantes de los dos grupos reciben el mismo curso de nivelación.
- Para la materia de Lenguajes de Programación I, en la cual se ha realizado el presente estudio, los contenidos que se dictan en la materia son idénticos, el docente es el mismo y su metodología de trabajo es igual para ambos grupos.
- Se ha tenido especial cuidado que el periodo estacional del año sea el mismo de marzo – agosto en los dos grupos de estudio.

Con estas condiciones se ha querido lograr que los factores exógenos tengan una baja probabilidad de afectación a los resultados del estudio, garantizando que la diferencia entre los dos grupos sea únicamente el método utilizado para la ejecución y seguimiento de los proyectos de fin de curso.

En el Periodo 2014 se ha utilizado un esquema tradicional para el seguimiento de los proyectos de fin de curso, en el cuál, el docente envía el proyecto, los estudiantes definen el tiempo que dedican al proyecto y el ritmo de avance. Siempre tienen a disposición al profesor en horas de consulta para resolver sus inquietudes. El docente percibe el avance individual y grupal sobre el proyecto con revisiones realizadas en sus horas de clase. El proyecto se presenta en un día específico al final del semestre a través de una exposición que incluye diapositivas, una exposición oral y la presentación del programa computacional.

En el Periodo 2015 se ha utilizado el método propuesto, aplicando su procedimiento y utilizando cada uno de sus artefactos. El método ha sido aplicado por un total de veinte y dos (22) días, periodo total de ejecución del proyecto de fin de curso.

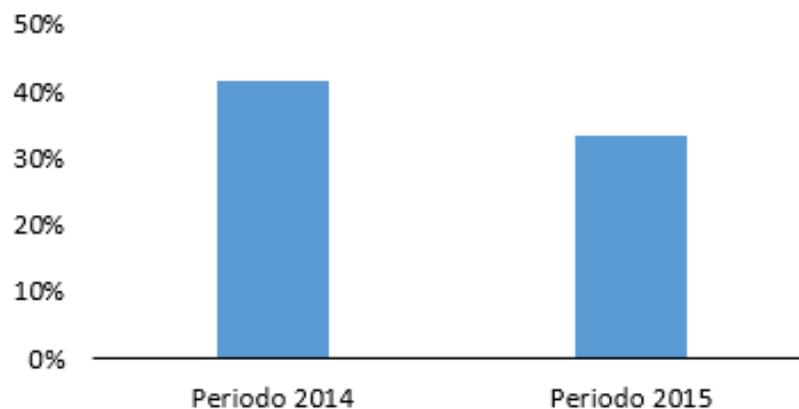
Para el experimento, un proyecto ha sido enviado a todos los equipos del Periodo 2014 y otro proyecto diferente ha sido enviado a todos los equipos del Periodo 2015. Esta diferencia en los proyectos ha sido intencional para evitar que los estudiantes del Periodo 2015 refieran o copien los proyectos del año anterior y se incurra con ello en un error del experimento. Sin embargo los proyectos para

los dos periodos, han sido definidos para presentar similares características y complejidad en sus requerimientos de implementación.

El instrumento de evaluación utilizado para calcular la calidad de los proyectos de fin de curso en los dos grupos, es la rúbrica de evaluación que representa a su vez el ARTEFACTO 07: Rúbrica de evaluación del método propuesto.

## 5. Resultados

Una vez aplicado el método tradicional en el grupo de estudiantes del Periodo 2014 y aplicado el método propuesto en el grupo de estudiantes del Periodo 2015, los resultados son los siguientes:



**Figura 1.** Media del grado (%) de calidad alcanzado en los proyectos de los periodos 2014 y 2015.

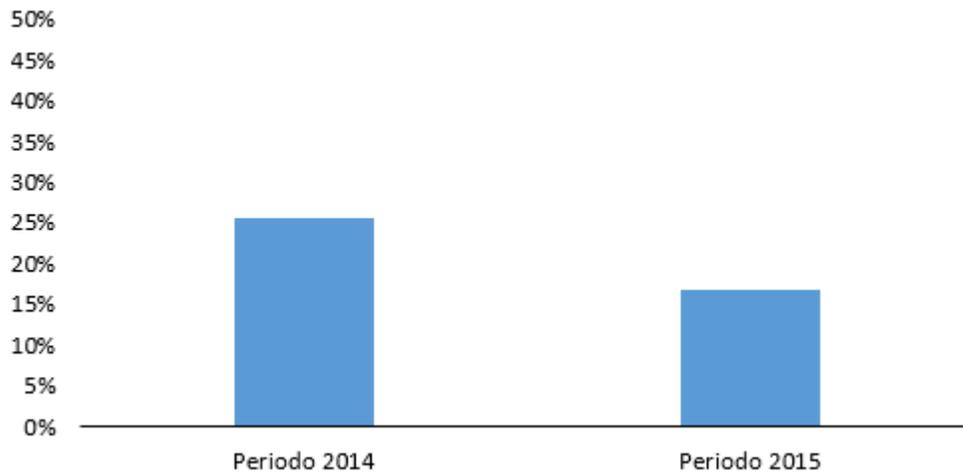
La Figura 1 muestra que el grado de calidad del Periodo 2014 con un valor del 41% es mayor que el ponderado para el Periodo 2015 que apenas alcanza un valor del 33%. Basado en estos valores cuantitativos se puede afirmar que la utilización del método propuesto en el Periodo 2015 no mejoró la calidad de los proyectos.

A primera vista esto haría suponer también que el método tradicional para el seguimiento y evaluación de proyectos es mejor que el propuesto, sin embargo, una lectura más detallada permite discutir y analizar que:

1. Para el Periodo 2014 la ponderación es subjetiva, debido a que se tiene menos información disponible para el docente. Las revisiones de aula y la presentación final son insuficientes para tener una percepción objetiva del trabajo individual y grupal de los estudiantes. La rúbrica de evaluación es llenada con base a esta subjetividad y puede ser sobreevaluada.
2. Para el Periodo 2015 el método propuesto brinda objetividad, ofreciendo al docente una trazabilidad completa del trabajo individual y grupal de sus estudiantes, no sólo en periodos cortos de aula o en la presentación final del proyecto sino durante todo el periodo de ejecución del proyecto. Esta visibilidad del trabajo de los estudiantes recae en una ponderación más acertada, más real, que evita sobrestimar el trabajo de estudiantes que no aporten con el proyecto, de equipos donde un estudiante sea el único ejecutor o de equipos que distribuyan ineficientemente su esfuerzo o su tiempo. La rúbrica de evaluación es llenada con base a toda esta evidencia y sus resultados pueden disminuir considerablemente respecto a una ponderación subjetiva.

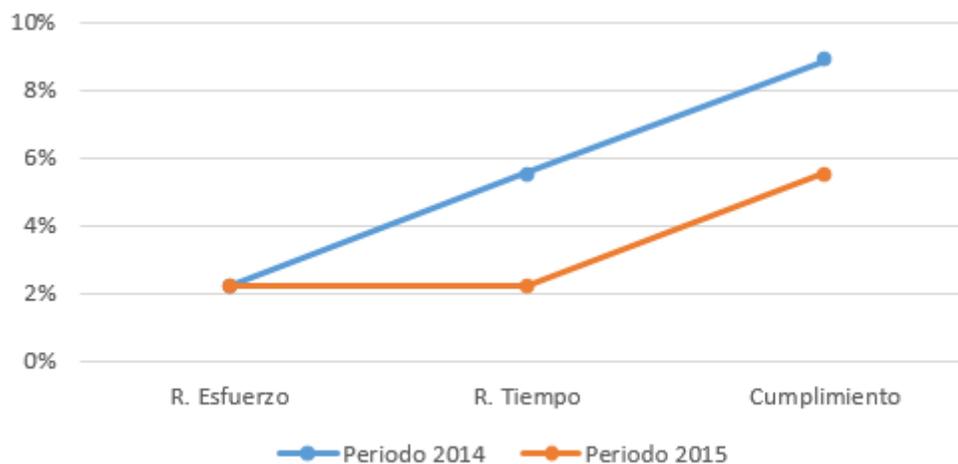
## **5.1 Componente grupal**

La rúbrica tiene componentes de evaluación tanto grupal como individual. La Figura 2 presenta la comparación entre los componentes grupales de los 2 periodos de estudio. Nótese que el valor máximo que pueden alcanzar las barras del gráfico es de 70%, porcentaje sugerido por MESEPP para este componente grupal.



**Figura 2.** *Componente grupal de calidad proyectos periodo 2014 y 2015*

En la Figura 2, al igual que en la Figura 1, se puede afirmar que el grado de calidad sigue siendo mejor para el Periodo 2014 con un valor del 26% por encima del Periodo 2015 con un valor del 17%. Y mientras estos valores cuantitativos permiten realizar esta afirmación, la diferencia de objetividad entre los dos periodos antes discutida, permite considerar el punto de vista que, los resultados en el Periodo 2015 simplemente evidencian con objetividad la ineficiencia del trabajo en equipo de los estudiantes.



**Figura 3.** *Line split considering angle  $\alpha$  for denoting the maximum deviation*

Desglosando el componente grupal, tal como lo presenta la Figura 3, se puede evidenciar que en los dos 2 periodos los estudiantes reparten ineficientemente el esfuerzo del proyecto con un valor ponderado del 2% sobre 20%. Esto significa que dentro de un mismo equipo hay estudiantes que hacen más esfuerzo y completan más cantidad de funcionalidad del proyecto, mientras que otros aportan poco o en ocasiones nada al proyecto.

La repartición del tiempo se ha ponderado mejor para el periodo 2014 con un valor de 6% sobre 20% con respecto al 2% del periodo 2015. Un equipo es mejor ponderado cuando realiza avances periódicos, progresivos y sostenidos del proyecto.

Se puede observar sin embargo que estas ponderaciones de la repartición del tiempo son bajas para los dos periodos, lo que da a notar que los estudiantes no reparten bien su tiempo, y como se analizará más adelante, tienden a realizar el mayor esfuerzo en los últimos días del periodo designado para la ejecución del proyecto.

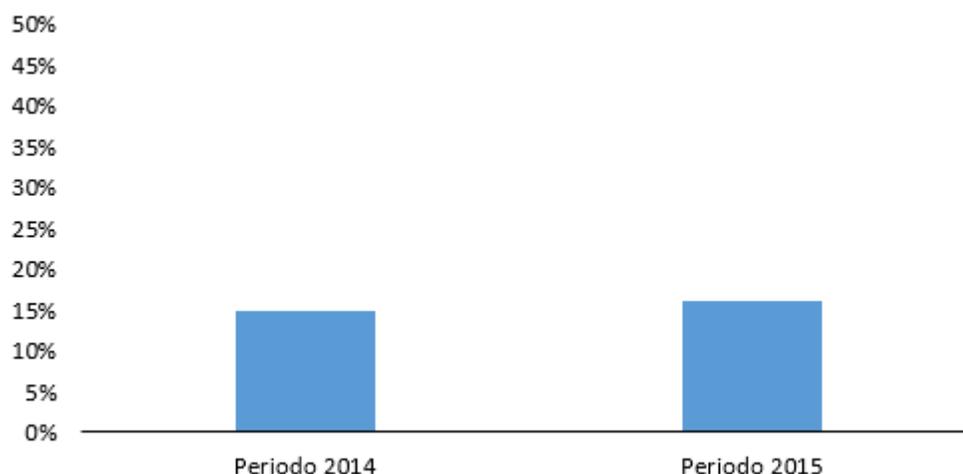
Y así como los índices de repartición de esfuerzo y de tiempo refieren la eficiencia con la que los estudiantes ejecutan el proyecto, el índice del cumplimiento presentado en la Figura 3 refiere la eficacia, referida como la capacidad de los estudiantes de cumplir con todos los requerimientos.

En esta valoración, la percepción de cumplimiento también fue mayor para el periodo 2014 con un valor del 9% sobre 20% contra el valor del 6% sobre 20% del periodo 2015. Nótese así mismo que estos valores son bajos, pues en ninguno de los 2 periodos los estudiantes, en promedio, cumplieron siquiera el 50% de los requerimientos solicitados para sus proyectos.

## **5.2 Componente individual**

La Figura 4 presenta la calidad de los proyectos en su componente individual, para los dos 2 periodos en estudio. En el gráfico el valor máximo de las barras

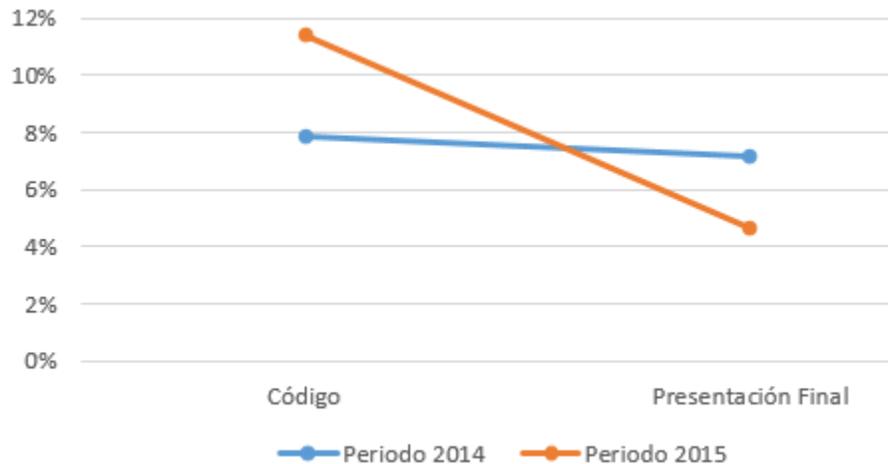
puede alcanzar el 30% que es el valor sugerido por MESEPP para este componente.



**Figura 4.** . *Componente individual de calidad de los proyectos periodo 2014 y 2015.*

Con la Figura 4 se puede afirmar que en lo que respecta al componente individual, es decir el trabajo por estudiante, el periodo 2015 presenta mejores resultados con un valor del 16% siendo un punto porcentual mayor respecto al 15% del periodo 2014.

Nótese sin embargo, que esto no significa que todos los estudiantes hayan hecho un mejor trabajo. La información detallada recogida en el periodo 2015, visibiliza que el trabajo destacado de unos pocos estudiantes ha elevado el valor de la media presentada en la figura.



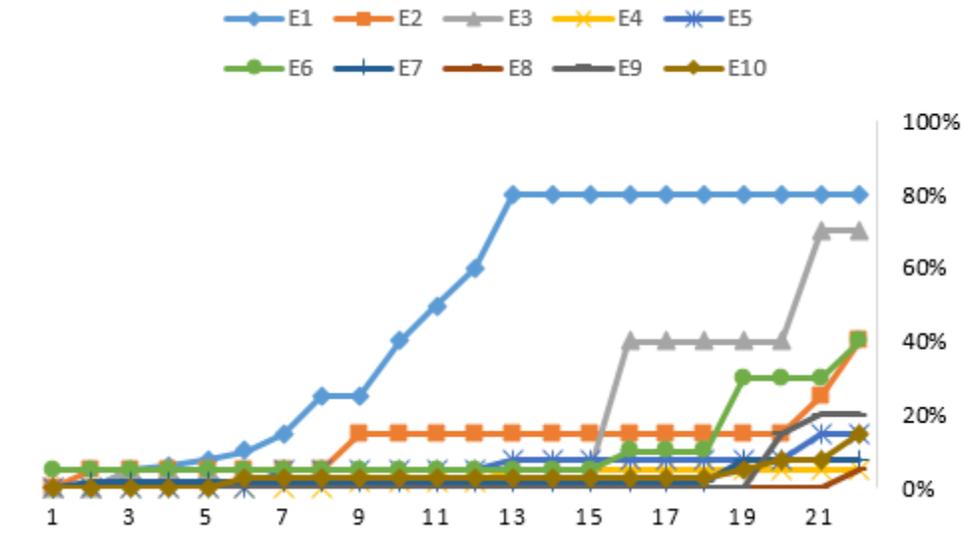
**Figura 5.** Índices del componente individual de calidad de proyectos periodo 2014 y 2015

Si desglosamos este componente individual, Como se presenta en el la figura 5, podemos notar que para el periodo 2015 la calidad del código es superior con un valor de 11% sobre 20% por encima del valor de 8% sobre 20% del periodo 2014. El índice de código indica que los estudiantes han utilizado los conceptos de programación enseñados en clase y los han utilizado bien.

Por otro lado el índice que pondera el desenvolvimiento individual de los estudiantes en su presentación final fue mejor para el periodo 2014 con un valor de 7% sobre 10% por encima del valor de 5% sobre 10% del periodo 2015.

### 5.3 Información enriquecida periodo 2015

Para el Periodo 2015, gracias a los datos registrados por el docente en el Artefacto 05 - Registro de avance, se puede obtener información adicional valiosa del avance del proyecto. La Figura 6 presenta una gráfica que evidencia el trabajo de los equipos (Ei) en el Periodo 2015.



**Figura 6.** Porcentaje total de requerimientos del proyecto cubiertos por los equipos (Ei) del día 1 al día 22 del periodo 2015

Observando el trabajo de todos los equipos, se puede decir que el trabajo del equipo 1 (E1) es un caso excepcional. Realiza un trabajo dedicado desde el día 1 hasta el día 13, día en el que completa ya el 80% del proyecto.

El equipo 3 (E3) realiza un trabajo incremental la última semana y en ella completa ya el 70% del proyecto.

El resto de equipos presentan un patrón de ejecución similar: no realizan ningún avance sino hasta los últimos días, en donde completan un bajo porcentaje del proyecto menor o igual al 40%.

Esta información evidencia los hábitos en la ejecución de los proyectos por parte de los estudiantes, donde las actividades se realizan en los últimos días acordados con el docente. En esos escasos días, el esfuerzo y el tiempo son insuficientes para cubrir con el 100% de los requerimientos exigidos por el proyecto.

Nótese sin embargo que la recolección de esta información adicional en el Periodo 2015 implica también la inversión de tiempo y esfuerzo adicional por parte del docente.

## 6. Conclusión

El estudio ha permitido evidenciar que la utilización del método MESEPP mejora el seguimiento de los proyectos de fin de curso, potenciando el proceso de educación, facilitando el trabajo colaborativo, permitiendo una evaluación objetiva, visibilizando los hábitos de estudio de los estudiantes, transparentando y automatizando las actividades inherentes a la ejecución de proyectos de software.

Para MESEPP GitHub es un componente fundamental. Con la herramienta, los estudiantes pueden mantener versionados sus proyectos y pueden potenciar su trabajo en equipo. El código aportado por cada estudiante, el número de aportes por día, la cantidad de líneas nuevas, editas y eliminadas por cada estudiante, son evidencias que se registran de manera automática con el uso de la herramienta.

Sin embargo la calidad de los aportes de los estudiantes, el avance porcentual día a día sobre el 100% de requerimientos del proyecto, así como la retroalimentación que debe ser entregada a cada estudiante, son todas ellas tareas que la plataforma GitHub desconoce y que no pueden ser automatizadas por ella, por tanto requieren del criterio, del tiempo y del esfuerzo del docente.

Ese sobre esfuerzo, de 1 a 3 horas diarias, que debe realizar el docente puede representar un limitante para el uso de un método como el sugerido en la realidad del aula. Sin embargo la creatividad del docente dueño de la materia para obtener colaboración de sus pares, de ayudantes de cátedra o de estudiantes de semestres superiores, podría solventar estas incomodidades intrínsecas en el nuevo método. Encontrar estas u otras estrategias en pos de la captura,

análisis y valoración basada en evidencias es fundamental para mejorar la calidad del producto y del proceso de ejecución de los proyectos de fin de curso.

The Protégé did not manifest any oxymoron regarding the ontological model we produced as a general constructor for virtual cities. The ontology presented in this paper has the potential to model up diverse kinds of cities. Nevertheless, regarding any unconsidered aspect in current ontology; it would be very easy to include additional categories or relations in the ontology.

Entre otras ventajas, el método MESEPP ha permitido visibilizar los hábitos de estudio y el verdadero trabajo que realizan los estudiantes para la ejecución de los proyectos. Se evidencia un trabajo sin una distribución equitativa entre los miembros del equipo y cuyo mayor esfuerzo se concentra en los días próximos a la finalización del tiempo fijado para su ejecución. Esta información progresiva y detallada recolectada con el uso de MESEPP permite al docente realizar una ponderación objetiva de sus estudiantes.

Debido a esta objetividad, las valoraciones de la calidad de los proyectos utilizando MESEPP han disminuido comparadas con las obtenidas con el método tradicional. Ocurre entonces que sin información y sin evidencias el docente tiende a sobrevalorar el esfuerzo individual y grupal de los estudiantes otorgándoles una ponderación mayor.

El componente de retroalimentación incluido en MESEPP ha tenido una respuesta positiva de los estudiantes, ellos han considerado las observaciones para mejorar su proyecto y corregir su camino a tiempo cuando la alternativa de solución escogida por ellos no ha sido la más adecuada. Con la ejecución de esta investigación se evidencia la importancia de un método como el propuesto. Si MESEPP no es el método más apropiado para un docente debido al sobre esfuerzo que implica su aplicación, de todas maneras el docente debería encontrar un método, que al igual que el sugerido, le entregue las evidencias suficientes para poder realizar un seguimiento y una valoración objetiva de los proyectos de fin de curso. Sin información y evidencia suficiente no se puede

medir bien, y sin una correcta medición no se puede mejorar la calidad del proceso y del producto final de la ejecución de los proyectos.

## Referencias

Bunge, M. (2005). *Buscar la filosofía en las ciencias sociales*. México: Siglo XXI Editores.

Chacon S, S. B. (2014). *Pro Git Second Edition*. Apress.

Coria, A., Pastor, I., & Torres, Z. (2013). Propuesta de metodología para elaborar una investigación científica en el área de Administración de Negocios. México: Revista científica Pensamiento y Gestión.

Git. (2015). *Git - Fast version control*. Recuperado el 2 de agosto de 2015, de Git: <https://git-scm.com/>

GitHub, Inc. (2015). *Twbs Bootstrap*. Recuperado el 10 de octubre de 2015, de <https://github.com/twbs/bootstrap>

Josh, D. (2014). *GitHub + University: How College Coding Assignments Should Work*. Recuperado el 13 de octubre de 2015, de [joshldavis.com/2014/01/19/github-university-how-college-assignments-should-work/](http://joshldavis.com/2014/01/19/github-university-how-college-assignments-should-work/)

Lawrance, J., Jung, S., & Wiseman, C. (2013). *Git on the Cloud in the Classroom*. 44th ACM technical symposium on Computer science education. Denver: ACM.

Ray, B., Posnett, D., Filkov, V., & Devanbu, P. (2014). *Large Scale Study of Programming Languages and Code Quality in Github*. 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. Hong Kong: ACM.

Real Academia Española. (2001). *Diccionario de la lengua española*. Recuperado el 25 de Julio de 2015, de Real Academia Española: <http://lema.rae.es/drae/?val=método>

Sawers, P. (11 de Febrero de 2014). *GitHub wants schools to collaborate on code*. (The next web, Inc.) Recuperado el 3 de octubre de 2015, de The Next Web: <http://thenextweb.com/insider/2014/02/11/github-wants-schools-collaborate-code/>

Shaffer, K. (26 de Mayo de 2013). *Push, Pull, Fork: GitHub for Academics*. Recuperado el 23 de abril de 2015, de HybridPedagogy:

<http://www.hybridpedagogy.com/journal/push-pull-fork-github-for-academics/>

Sierra, R. (2001). Técnicas de investigación social. Teoría y ejercicios. Madrid: S.A. EDICIONES PARANINFO.

Xu, Z. (2012). Using Git to Manage Capstone Software Projects. Seventh International Multi-Conference on Computing in the Global Information Technology. Venice, Italy: IARIA.

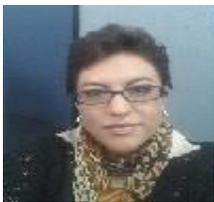
Yu, Y., Yin, G., Wang, H., & Wang, T. (2014). Exploring the Patterns of Social Behavior in GitHub. 1st International Workshop on Crowd-based Software Development Methods and Technologies. Hong Kong: ACM Press.

Zagalsky, A., Feliciano, J., Storey, M.-A., Zhao, Y., & Wang, W. (2015). The Emergence of GitHub as a Collaborative Platform for Education. 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. Vancouver, BC, Canada.

## Notas biográficas:



**Javier Salazar Zárate.** Ingeniero en Sistemas Informáticos; Magister en Informática Educativa; Arquitecto de Software. Consultor de tecnologías para desarrollo web. Fundador de eduKlip, una startup educativa de base tecnológica en Ecuador.



**Blanca Hidalgo Ponce.** Ingeniero en sistemas; Magister en informática aplicada y diplomada en manejo de información a través de internet en la ESPOCH. Coordinadora del programa de maestría en interconectividad de redes IPEC-ESPOCH. Miembro del grupo de investigación en la ingeniería de software GrII Soft –FIE-ESPOCH.



**Narcisa de Jesús Salazar** Doctora en Matemática; Master en Informática Aplicada; Docente de Cálculo Matemático y Probabilidades de la ESPOCH; Directora de las Maestrías en Informática Aplicada y de Matemática Básica; Vicedecana Facultad de Informática y Electrónica de la ESPOCH.



**Byron Vaca Barahona.** Doctor en Tecnologías Educativas: e-learning y gestión del conocimiento en la Universidad Roviera i Virgili (España). Miembro del grupo de investigación “Applied Research Group in Education and Technology (ARGET)”.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

# **Sistema de Navegación Reactiva Difusa para Giros Suaves de Plataformas Móviles Empleando el Kinect**

**Fuzzy Mobile Reactive Navigation System for Smooth Turns  
by Using Kinect**

Claudia Cruz Martínez<sup>1</sup>  
ccmartinez@inaoep.mx

Ignacio Algreto Badillo<sup>2</sup>  
ignacio.algreto@uptlax.edu.mx

J. Jesús Arellano Pimentel<sup>3</sup>  
jjap@sandunga.unistmo.edu.mx

Ernesto Cortés Pérez<sup>3</sup>  
neto\_144@sandunga.unistmo.edu.mx

Francisco Aguilar Acevedo<sup>3</sup>  
aguilar.afco@sandunga.unistmo.edu.mx

Luis Alberto Morales Rosales<sup>4</sup>  
lamorales@itsm.edu.mx

<sup>1</sup> Instituto Nacional de Astrofísica, Óptica y  
Electrónica, México

<sup>2</sup> Universidad Politécnica de Tlaxcala, México

<sup>3</sup> Universidad del Istmo, México

<sup>4</sup> Instituto Tecnológico Superior de Misantla, México

**Resumen:** La navegación en un robot móvil es la habilidad para desplazarse de un lugar a otro dentro de un entorno evitando los obstáculos que se presenten. La autonomía de un robot móvil se basa en su sistema de navegación. La aplicación de técnicas de Inteligencia Artificial como la lógica difusa y el uso de la visión por computadora son dos herramientas empleadas para cumplir esta tarea. En algunos sistemas de navegación la seguridad y la facilidad de operación son factores muy importantes. En estos casos, la tolerancia a la incertidumbre de información, la reacción ante objetos imprevistos, y la navegación mediante giros suaves son argumentos del diseño de estos sistemas. En este artículo se presenta el desarrollo de un sistema de navegación reactiva difusa que emplea los datos de profundidad del sensor Kinect, algoritmos de visión por computadora, y lógica difusa, para generar ángulos de giro suave para la navegación de un robot móvil. En pruebas realizadas con la plataforma móvil ERA-MOBI se observaron giros suaves con un porcentaje de evasión de obstáculos del 85.7%.

**Palabras clave:** navegación reactiva, lógica difusa, Kinect, profundidad promedio, robot móvil.

**Abstract:** Navigation on a mobile robot is the ability to move from one place to another within an environment avoiding obstacles that arise. The autonomy of a mobile robot is based on your navigation system. The application of Artificial Intelligence techniques such as fuzzy logic and using computer vision are two tools used to accomplish this task. In some navigation systems the safety and ease of operation are important factors. In these cases, the tolerance for uncertainty information, the reaction to unforeseen objects, and navigation through smooth turn are design arguments of these systems. This article describes the development of a system of fuzzy logic-based reactive navigation using data depth of Kinect sensor, computer vision algorithms, and fuzzy logic, to

generate smooth turn angles for navigation of a mobile robot. In tests with the platform ERA- MOBI smooth turns with a percentage of obstacle avoidance of 85.7 % were observed.

**Keywords:** reactive navigation, fuzzy logic, average depth, mobile robot.

# 1. Introducción

A diferencia de la mayoría de robots industriales con un limitado espacio de trabajo, los robots móviles pueden moverse libremente para alcanzar sus objetivos deseados (Tzafestas, 2014). Esta capacidad de movilidad hace a estos robots adecuados para un gran repertorio de aplicaciones en entornos estructurados y no estructurados. Los robots móviles que emplean ruedas para su locomoción asumen diferentes configuraciones, las cuales les confieren diferentes características y propiedades respecto a su eficiencia energética, dimensiones, cargas útiles y maniobrabilidad. Dentro estas configuraciones la diferencial es la más frecuente en robots para interiores (Ollero, 2001).

Por otra parte, un robot móvil se enfrenta a entornos locales diferentes cada vez que se mueve, teniendo la capacidad de influir y ser influido por estos (Kelly, 2013). Más importante, el mundo es un lugar peligroso, y que a menudo no puede ser diseñado para adaptarse a las limitaciones del robot, por lo que la movilidad aumenta el nivel de inteligencia necesaria. El éxito al hacer frente a las diferentes demandas y riesgos de cada lugar y cada situación es un reto importante incluso para los sistemas biológicos.

En este aspecto, la navegación es una metodología que permite guiar a un robot móvil de manera segura y estable, a través de un entorno con obstáculos con el fin de ejecutar alguna tarea en específico (Murphy, 2000). Durante las últimas décadas se han realizado importantes esfuerzos en la aplicación de técnicas de Inteligencia Artificial a la navegación (Ollero, 2001). La lógica difusa es uno de los métodos empleados que proporciona una tolerancia inherente a la incertidumbre en los datos (Yerubandi, Reddy, & Kumar, 2015). Además, cuando

se selecciona un sistema de inferencia difuso de forma adecuada es posible modelar eficazmente la experiencia humana en una aplicación específica (Mohan, 2005).

Desde el punto de vista de la planificación, existen diferentes arquitecturas diseñadas teniendo en cuenta especificaciones como el tiempo de respuesta del sistema y la disponibilidad de información. La solución se sitúa normalmente entre dos extremos: la planificación puramente estratégica y la puramente reactiva. En la planificación reactiva se supone que el entorno es incierto, buscándose la mayor flexibilidad posible para reaccionar en cualquier instante lo suficientemente rápido a las discrepancias entre el modelo actual y la realidad observada en el entorno (Ollero, 2001). Un ejemplo de planificación reactiva se presenta en Suárez y Loaiza (2015), con la integración de un robot móvil y un sistema de visión artificial, para detectar y evadir obstáculos presentes en la trayectoria del robot, empleando una estrategia de navegación de tipo local, basada en campos de potencial.

Algunas aplicaciones de los robot móviles requieren de sistemas de navegación donde la seguridad y la facilidad de operación son los factores más importantes de evaluación, tal es el caso de las sillas de ruedas robotizadas (Kuno, Shimada, & Shirai, 2003). Proporcionar autonomía es una forma de mejorar ambos factores. En este tipo de aplicación resulta conveniente evitar ángulos de giro bruscos que puedan representar un riesgo de lesiones en el usuario, siendo una alternativa de solución un incremento gradual no pronunciado en el ángulo de giro al eludir un obstáculo, es decir un ángulo de giro suave. La evasión de obstáculos empleando sensores infrarrojos, ultrasónicos, de visión, y otros sensores son también motivo de estudio en estos sistemas. En este sentido uno de los sensores utilizados en aplicaciones robóticas, en particular en tareas de navegación de interiores es el sensor Kinect de Microsoft® , el cual utiliza la tecnología de una cámara RGB y una cámara infrarroja para diferenciar profundidad (El-laithy, Huang, & Yeh, 2012). En Mittal y Goyal (2014) se presenta un sistema de navegación para una silla de ruedas que emplea un sistema

operativo Linux, sensores infrarrojos y una cámara Kinect, para la evasión de obstáculos, el paso en puertas y el seguimiento de paredes.

Así, en este artículo se presenta el diseño e implementación de un sistema de navegación reactiva difusa, que a partir de los datos de profundidad obtenidos con el sensor Kinect, defina ángulos de giro suave para la navegación de una plataforma/robot móvil. El sistema considera la incertidumbre de información como argumento de su diseño.

## 2. Sistemas de navegación reactiva difusa

Los sistemas de navegación reactiva basados en control difuso, difieren unos de otros en los algoritmos de procesamiento de información, tipo de sistema de inferencia, el conjunto de reglas, las funciones de membresía, variables de entrada y salida, o en el tipo de sensor que utiliza para la adquisición de información del ambiente.

En algunos sistemas de navegación difusa se plantea el uso de imágenes digitales 2D. Por ejemplo, en Ramírez, Gómez, Martínez, y López (2011) se emplea un sistema de inferencia Mamdani, con funciones de membresía tipo triangular y trapezoidal, los cuales permiten ángulos de giro suaves, sin embargo la detección y evasión de obstáculos en curso de colisión o imprevistos no fue considerada, limitándose únicamente a ambientes tipo pasillo lineal.

Algunos trabajos (Mester, 2010; Bueno & Ríos, 2008; Villaseñor et al., 2010; Parra, Ríos, & Bueno, 2007) emplean sensores ultrasónicos y un control difuso con sistemas de inferencia tipo Mamdani; predominan las funciones de membresía tipo triangular, algunas veces en combinación con funciones de membresía tipo gaussiana y trapezoidal; estos trabajos emplean una navegación reactiva a base de comportamientos. Por su parte Acosta, Gallardo, y Pérez (2016) presentan una arquitectura de control basada en el paradigma reactivo,

para la navegación autónoma de robots móviles con capacidades de cómputo limitadas haciendo uso de la teoría de control difuso en una plataforma robótica provista de tres sensores ultrasónicos dedicada al seguimiento de paredes. Sin embargo, el reducido alcance de los sensores ultrasónicos empleados en estos trabajos impide prever colisiones futuras. Además, los giros suaves tampoco son argumentos de dichos trabajos.

Based on the previous statement, despite the vast quantity of tools to help urban modeling, there is a lack of tools devoted to simulate study-cases focusing in city operation, instead of its mere visualization; since modern tools frequently focus in 3D simulation and do not have enough power to deal with issues regarding how chaotic and erratic a real city is.

En Aguirre, Gómez, Muñoz, y Ruiz (2003); Cupertino, Giordano, Naso y Delfine (2006); y Chica, Pineda, y Esmeral (2009) se proponen arquitecturas híbridas de fusión sensorial entre sensores ultrasónicos o infrarrojos con imágenes 2D. Sin embargo, el procesamiento de las imágenes 2D y la fusión sensorial conllevan un alto costo computacional. Por su parte en Cupertino et al. (2006) y Chica et al. (2009) se utilizan funciones de membresía triangulares. En cualquiera de los casos no se hace alusión a los giros suaves como elemento de la navegación, ni se considera la prevención de colisiones futuras.

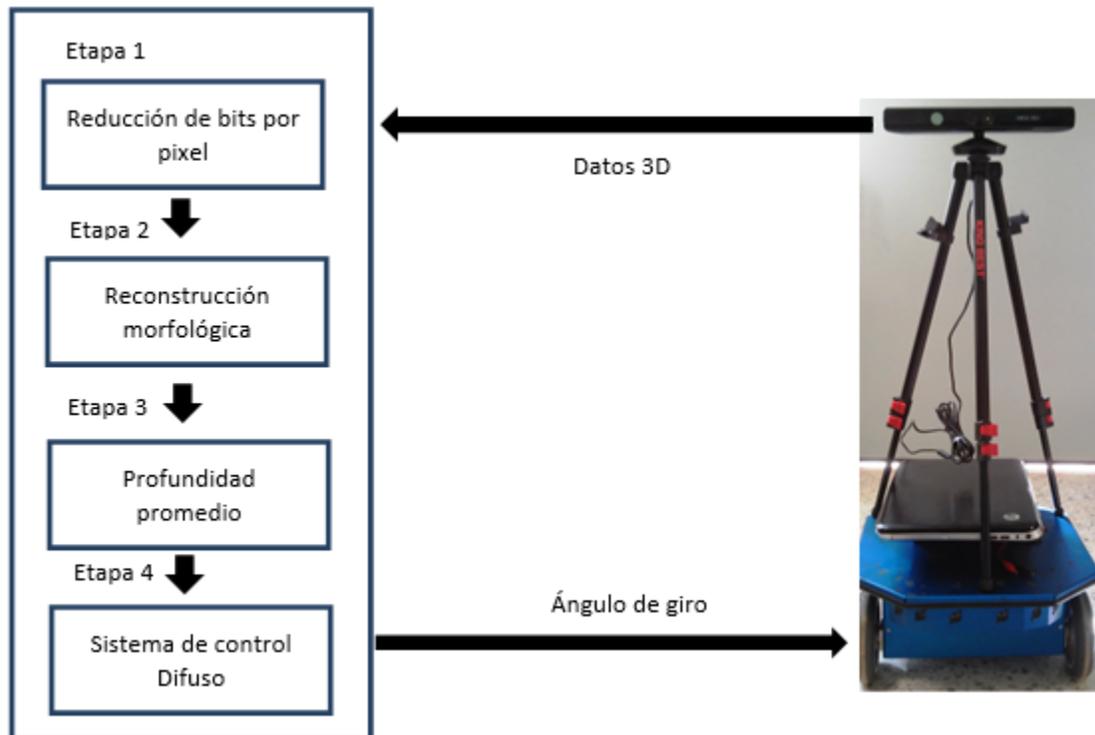
Otros trabajos proponen sistemas de navegación utilizando el Kinect como sensor principal y la lógica difusa para el procesamiento de la información. En Díaz y Romero (2012) se aplican un filtro Gaussiano a la imagen de profundidad, para posteriormente localizar el centroide de un objeto próximo. Se obtiene una distancia promedio para evadir el obstáculo y se gira a lado contrario de la posición del objeto, deteniendo motores o avanzando hacia delante. Csaba y Vámosy (2012) implementan algoritmos para planificar la ruta y evadir obstáculos, usando un sistema de inferencia Mamdani de 16 reglas, funciones de membresía trapezoidal y triangular, tres variables de entrada (profundidad del objeto más cercano de lado derecho, del lado izquierdo y al frente), y una

variable difusa de salida (ángulo de giro del robot móvil). Benavidez y Jamshidi (2011) emplean las imágenes de profundidad del Kinect junto con sensores odométricos. El control difuso utiliza funciones de membresía de tipo triangular. A las imágenes de profundidad se les aplica un filtro Gradiente-Log para identificar áreas transversales (suelos). Estos trabajos basados en el Kinect tienen como característica común que buscan evadir obstáculos en un determinado rango de profundidades, sin considerar lo pronunciado de los giros ni las colisiones futuras.

Los trabajos mencionados están orientados hacia la navegación reactiva y tienen en común aplicar la lógica difusa para determinar el siguiente movimiento a ejecutar por el robot móvil. Utilizan información de profundidad o proximidad para detectar los obstáculos, evadirlos, e incluso extraer las dimensiones, las formas y centroide de los objetos.

### 3. Propuesta de sistema de navegación para giros suaves

En la Figura 1 se muestra un esquema del sistema de navegación propuesto. El Kinect provee los datos de profundidad los cuales son procesados en una computadora en cuatro etapas para generar finalmente el ángulo de giro suave evitando colisiones futuras e imprevistas. La plataforma móvil de prueba fue el robot con ruedas de tipo diferencial ERA-MOBI, el cual se encuentra equipado con una computadora a bordo y una antena Wifi (Videre Design, 2009). El ángulo de giro es enviado al robot vía sockets empleando el servidor Player/Stage. Player es un popular servidor genérico de código abierto (open-source) empleando en robótica para el control de sensores y actuadores (Amanda, 2010). El robot ERA-MOBI es compatible con una amplia gama de sensores, incluyendo infrarrojos, sonares, telémetros láser y de visión estereoscópica. En los siguientes apartados se describen a detalle cada una de las cuatro etapas que transforman los datos de profundidad del Kinect en un ángulo de giro suave.



**Figura 1.** Esquema general del sistema de navegación reactiva difusa, etapa 1: Reducción de bits por pixel, etapa 2: Reconstrucción morfológica, etapa 3: Profundidad promedio, y etapa 4: Sistema de control Difuso.

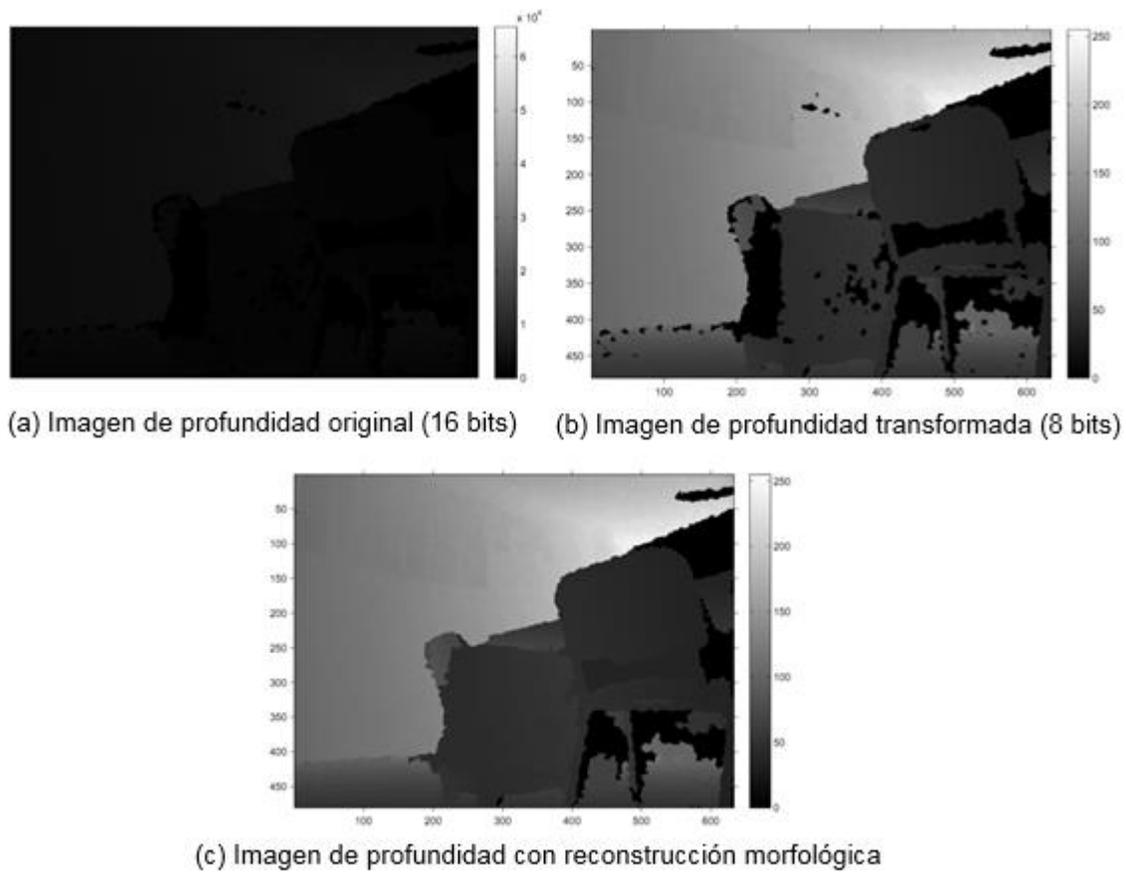
### 3.1 Reducción de bits por pixel

La información de profundidad se adquiere mediante el Kinect en una imagen de 640x480, donde cada pixel indica una cantidad de profundidad en una escala de 0 milímetros (color negro) hasta 65,535 milímetros (color blanco), tal como se puede apreciar en la Figura 2a. En esta imagen la mayoría de los pixeles caen dentro del espectro más oscuro ya que superan el valor máximo de visión real del Kinect (10,000 milímetros). El proceso de reducción de bits por pixel transforma la imagen de 16 bits entregada por el Kinect en una de 8 bits (ver Figura 2b), donde el valor 0 representa objetos muy cercanos, objetos muy lejanos u objetos que no fueron identificados. Esto conduce a que el sistema de navegación busque mantener una dirección hacia lugares donde exista suficiente información de profundidad, evitando dirigirse hacia zonas donde

predominen valores de 0, esta característica ayuda a evitar posibles colisiones futuras.

### 3.2 Reconstrucción morfológica

La morfología matemática sobre imágenes digitales permite simplificar los datos de una imagen, conservar las características esenciales y eliminar los aspectos que sean irrelevantes (Parker, 2010). En este caso el proceso de reconstrucción morfológica busca conservar las características de profundidad, y a su vez eliminar pequeñas zonas desconocidas rodeadas por zonas conocidas, brindando así una mayor definición de los objetos y sus profundidades en el ambiente. Esto se logra a través del algoritmo de erosión geodésica (Vincent, 1993), realzando los píxeles con intensidades mínimas siempre y cuando sus píxeles vecinos sean de intensidades mayores, dando como resultado una imagen de profundidad con un menor número de partículas oscuras. En la Figura 2c se muestra un ejemplo de los resultados de la reconstrucción morfológica.



**Figura 2.** Reducción de bits por pixel y reconstrucción morfológica.

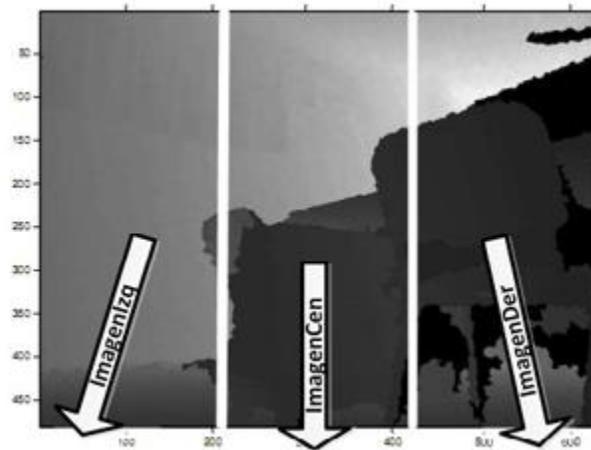
### 3.3 Profundidad promedio

Un método usado para el tratamiento de la imagen de profundidad es dividir esta en sub-imágenes (Csaba y Vámosy, 2012; Csaba, 2013). En el sistema propuesto la imagen es dividida en tres secciones de 211x480 píxeles, las cuales representan las direcciones posibles en las que la plataforma móvil puede moverse: girar a la derecha, seguir de frente y girar a la izquierda (véase Figura 3).

El sistema emplea el método planteado por Otsu (1979), para calcular el umbral que permita diferenciar las profundidades en valores mayores y menores. Para cada sub-imagen, si la cantidad de datos desconocidos (ceros) es mayor a un determinado porcentaje (67% para este caso), entonces los datos en la sub-

imagen indicarán obstáculos muy cercanos, y la profundidad promedio se calculará de la media de las profundidades menores. En caso contrario, se calcula la profundidad promedio de la media de las profundidades mayores. Si en la escena no existen profundidades menores, entonces se consideran los valores de profundidad mayores.

Mediante el cálculo de la profundidad promedio en cada sub-imagen se conoce que región ofrece una menor probabilidad de colisión. A partir de esta información, el sistema de navegación reactiva debe calcular la dirección de giro, considerando los datos imprecisos, y evitando movimientos bruscos. Bajo este escenario se plantea el uso de un control difuso.



**Figura 3.** Sub-imágenes de profundidad.

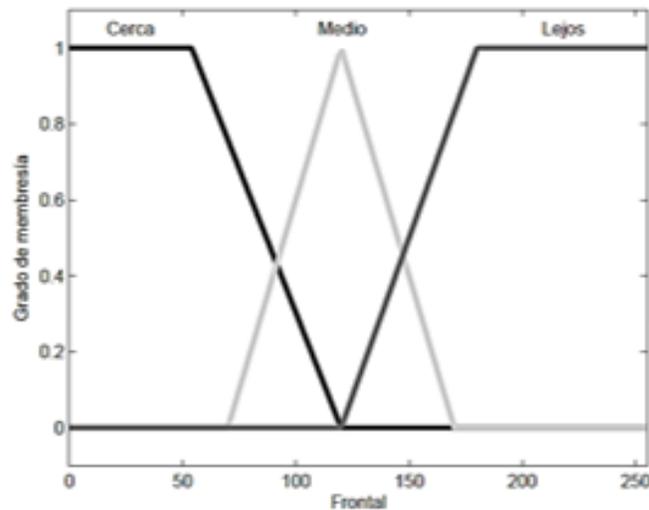
### 3.4 Sistema de control difuso

Esta etapa consiste en la implementación de un control difuso para la navegación reactiva de la plataforma móvil. Se empleó la herramienta FuzzyLogicToolbox™ de Matlab®, para este desarrollo.

El sistema de inferencia difuso recibe como entrada los valores de profundidad promedio obtenidos en cada sub-imagen, generando como salida una variable en grados sexagesimales que va desde  $-90^\circ$  a  $90^\circ$ , el cual representa el ángulo de giro. Se utiliza un sistema de inferencia Sugeno, el cual es

computacionalmente efectivo al generar salidas numéricas que no requieren un proceso de defusificación. Los sistemas Sugeno trabajan correctamente con técnicas adaptativas y de optimización (Negnevitsky, 2005; Jang, Sun, & Mizutani, 1997). Además, como se sugiere en Kornuta & Marinelli (2013), su uso en sistemas de navegación garantiza la continuidad de la superficie de salida.

Para cada una de las variables de entrada Izquierda, Frontal y Derecha se definen 3 valores lingüísticos: Cerca representando las profundidades de los “objetos cercanos”, Medio considerando la profundidad de aquellos objetos que no están “ni muy lejos ni muy cerca”, y Lejos para indicar las profundidades de los “objetos alejados” del Kinect. Se hacen uso de funciones de membresía trapezoidales para los valores lingüísticos Cerca y Lejos, y triangular para Medio (véase Figura 4). Este tipo de funciones de membresía no representan operaciones complejas como es el caso de la función gaussiana o la campana generalizada (Jang et al., 1997).



**Figura 4.** Variable de entrada Frontal: Cerca, Medio, Lejos.

La variable de salida Ángulo se divide en siete valores lingüísticos para interpretar qué tan pronunciado es el giro para navegar. La función para la variable lingüística de salida se define mediante la Ecuación 1.

$$S_i = A_i x + B_i y + C_i z + D_i$$

donde: •  $S_i$  : es la variable de salida para cada valor lingüístico  $i$ . •  $A_i, B_i, C_i, D_i$ : son valores constantes para cada valor lingüístico  $i$ . •  $x, y, z$ : son las variables de entrada.

Para evitar un comportamiento complejo se utilizó un sistema Sugeno de orden cero (Jang et al., 1997), por lo que la función para cada variable lingüística se define solo en términos de una constante  $D_i$ . Empleado como referencia los trabajos realizados por Csada y Vámosy (2012) y Csada (2013), para cada valor lingüístico se definieron las siguientes funciones:  $D_1 = 90$  para MuyPositivo,  $D_2 = 60$  para Positivo,  $D_3 = 30$ , para PocoPositivo,  $D_4 = 0$ , para Cero,  $D_5 = -30$ , para PocoNegativo,  $D_6 = -60$ , para Negativo, y  $D_7 = -90$ , para MuyNegativo.

Dependiendo del número de permutaciones de los valores lingüísticos (Cerca, Medio y Lejos) con respecto a las variables de entrada (Izquierda, Frontal y Derecha), se desarrolla un conjunto de reglas difusas para obtener el valor de salida que corresponde al giro de la plataforma móvil. El control difuso emplea 27 reglas divididas en cuatro grupos dependiendo de un objetivo en común: reglas para la ejecución de giros pronunciados, reglas para la predicción de colisiones, reglas para el movimiento en línea recta, y reglas para giros básicos. En la Tabla 1 se muestran las reglas difusas planteadas. Las reglas buscan emular la reacción humana durante la navegación, ya que generan reacciones graduales que van desde prácticamente no ejecutar giro hasta los movimientos bruscos ante colisiones imprevistas o en movimiento. El conjunto de reglas tiene como principal objetivo producir giros suaves que permitan asegurar la integridad de la plataforma móvil.

		SI			ENTONCES
		<i>Izquierda</i>	<i>Frontal</i>	<i>Derecha</i>	<i>Ángulo</i>
Conjunto de reglas para giros pronunciados	1	Cerca	Cerca	Cerca	<u>MuyNegativo</u>
	2	Cerca	Cerca	Medio	<u>MuyPositivo</u>
	3	Cerca	Cerca	Lejos	<u>MuyPositivo</u>
	4	Medio	Cerca	Cerca	<u>MuyNegativo</u>
	5	Medio	Cerca	Medio	<u>MuyNegativo</u>
	6	Medio	Cerca	Lejos	<u>MuyPositivo</u>
	7	Lejos	Cerca	Cerca	<u>MuyNegativo</u>
	8	Lejos	Cerca	Medio	<u>MuyNegativo</u>
	9	Lejos	Cerca	Lejos	<u>MuyNegativo</u>
Conjunto de reglas para predecir colisiones	10	Cerca	Medio	Medio	<u>PocoPositivo</u>
	11	Medio	Medio	Cerca	<u>PocoNegativo</u>
	12	Medio	Medio	Lejos	<u>PocoPositivo</u>
	13	Lejos	Medio	Medio	<u>PocoNegativo</u>
	14	Lejos	Medio	Lejos	<u>PocoNegativo</u>
	15	Cerca	Lejos	Lejos	<u>PocoPositivo</u>
	16	Medio	Lejos	Lejos	<u>PocoPositivo</u>
	17	Lejos	Lejos	Cerca	<u>PocoNegativo</u>
	18	Lejos	Lejos	Medio	<u>PocoNegativo</u>
Conjunto de reglas para el movimiento en línea recta	19	Cerca	Medio	Cerca	Cero
	20	Medio	Medio	Medio	Cero
	21	Cerca	Lejos	Cerca	Cero
	22	Cerca	Lejos	Medio	Cero
	23	Medio	Lejos	Cerca	Cero
	24	Medio	Lejos	Medio	Cero
Conjunto de reglas para giros básicos	25	Lejos	Lejos	Lejos	Cero
	26	Cerca	Medio	Lejos	Positivo
	27	Lejos	Medio	Cerca	Negativo

Tabla 1. Reglas difusas.

## 4. Resultados

Las pruebas al sistema de navegación reactiva difusa se ejecutaron en dos entornos diferentes: doméstico y laboratorio de cómputo. En ambos casos, el Kinect se colocó a un metro de altura (con una inclinación vertical de 0°) respecto a la plataforma móvil. El sistema de control difuso trabaja con las imágenes de profundidad, empleando las imágenes RGB únicamente para una visualización

más clara del ambiente. La aplicación fue ejecutada en dos equipos de cómputo diferentes, cuyas características se muestran en la Tabla 2.

Tipo de Equipo	Marca	Sistema Operativo	Procesador	Velocidad del Procesador	Memoria RAM
Laptop	Toshiba Satellite L855	Windows 7	Intel Core i7	2.30 GHz	8 GB
PC	Dell Vostro 420	Windows XP	Intel Core 2 Duo	3.16 GHz	3 GB

**Tabla 2. Características de los equipos de cómputo para pruebas.**

Se realizaron 1000 experimentos para calcular los tiempos de ejecución del sistema propuesto. En la Tabla 3 se observan el tiempo de captura de la imagen, los tiempos de ejecución de cada una de las cuatro etapas, y el tiempo total.

Tipo de Equipo	Captura de imagen	Normalización de los datos	Reconstrucción morfológica	Profundidad promedio	Control difuso	Tiempo total
Laptop	10.4 ms	3.4 ms	4.1 ms	5.4 ms	15.6 ms	38.9 ms
PC	6.4 ms	6.7 ms	4.8 ms	6.7 ms	18.9 ms	43.5 ms

**Tabla 3. Tiempos de ejecución promedio.**

El cálculo del ángulo de giro mediante el controlador difuso supera el tiempo de las demás etapas indicando un mayor consumo de recursos computacionales. Se observa que los procesos con el menor tiempo promedio varían entre equipos de cómputo. En el caso de la Laptop, el proceso de normalización de datos de profundidad consume el menor tiempo con 3.4 ms. En la PC el proceso de reconstrucción morfológica genera el menor tiempo con 4.8 ms.

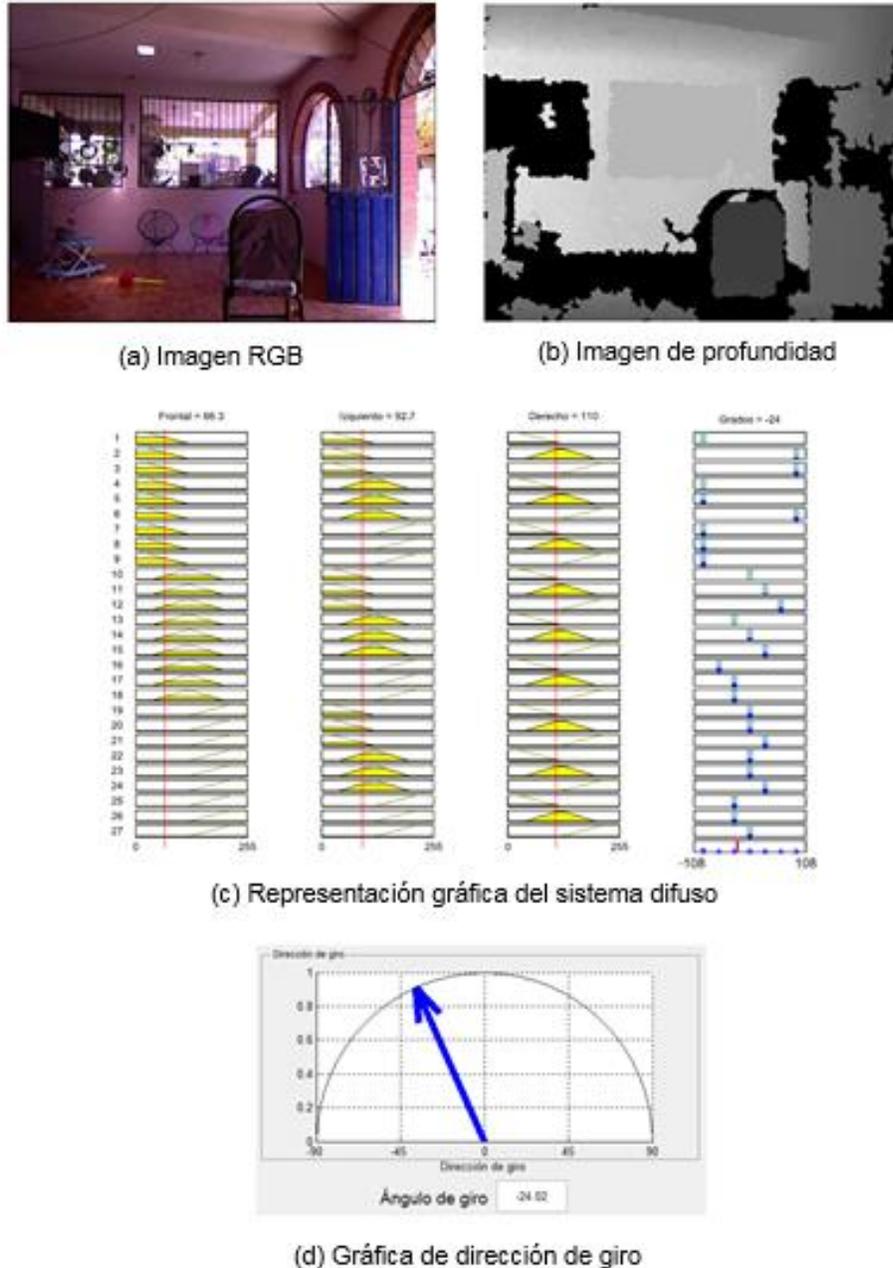
El sistema de navegación reactiva difusa ejecuta sus operaciones en un rango de entre 38.9 ms y 43.5 ms. Si se considera una frecuencia de procesamiento (a nivel de hardware) del Kinect de 30 Hz (Webb & Ashley, 2012; Cruz, Lucio, & Velho, 2012; Andersen et al., 2012), lo cual representa un tiempo aproximado de 33.33 ms entre capturas, es posible valorar una adecuada sincronía entre el procesamiento del Kinect y el sistema de navegación.

Durante las primeras pruebas descritas, el Kinect no fue colocado directamente sobre el robot móvil, lo cual conllevó a que el ángulo de giro resultante fuera ejecutado un operador humano. Posteriormente se realizaron pruebas usando el robot ERA-MOBI y la computadora Laptop, realizando el montaje mostrado en la Figura 1. Se programó un cliente en la Laptop y una aplicación cliente/servidor Player en la computadora a bordo del robot para comunicar y ejecutar el ángulo de giro en el robot móvil. En las pruebas realizadas el sistema acertó en las decisiones de dirección de giro evitando obstáculos en un 85.7%.

En la Figura 5 se ilustra el comportamiento del sistema dentro de un ambiente doméstico sin movimiento de los objetos e iluminación natural moderada. En las Figuras 5a y 5b se presentan la imagen RGB y su correspondiente imagen de profundidad. En la Figura 5c se muestra una representación del sistema difuso, el cual identifica las profundidades en cada lado de la imagen otorgándoles valores lingüísticos usando las reglas difusas, para finalmente a través del método de inferencia genere el ángulo de giro en grados (véase Figura 5d). Se observó que el sistema trata de seguir el espacio con la menor probabilidad de colisión manteniéndose estable al continuar en espacios con información conocida. Dependiendo de la cantidad de datos desconocidos, el sistema infiere si debe ignorarlos o considerarlos dentro del cálculo de la profundidad promedio por cada sub-imagen. En el ejemplo de la Figura 5, los datos de profundidad desconocidos son ignorados al no abarcar un área mayor a dos terceras partes de la imagen total.

En la Figura 6 se muestra un caso donde las profundidades van desde los rangos cercanos a los lejanos en un ambiente en donde un objeto se va acercando poco a poco al sensor (leves movimientos). En la Figura 6a se presenta la imagen RGB. En la Figura 6b se muestra la imagen de profundidad, en donde se observa cómo algunos elementos toman valores desconocidos. El sensor adquiere la imagen de profundidad para que así el sistema procese la información y reaccione para evadir al objeto en movimiento (bebé en andadera en la Figura 6a). Es de notar que las partes desconocidas no son consideradas por el

sistema, ya que a pesar de abarcar un área casi uniforme, no representan un objeto. El sistema define un giro hacia la izquierda de  $-47.3^\circ$  (véase Figura 6c) al observar una profundidad promedio mayor (193) en la parte izquierda (véase Figura 6d).



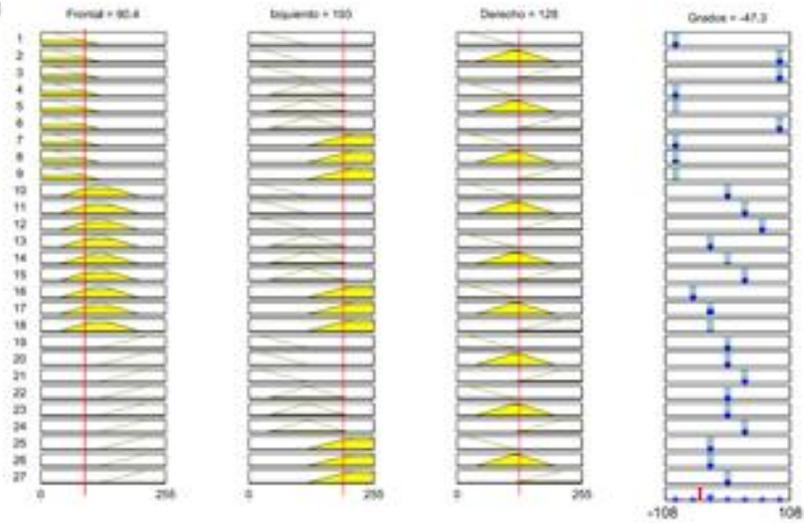
**Figura 5.** Comportamiento del sistema en ambientes domésticos, sin movimiento de los objetos



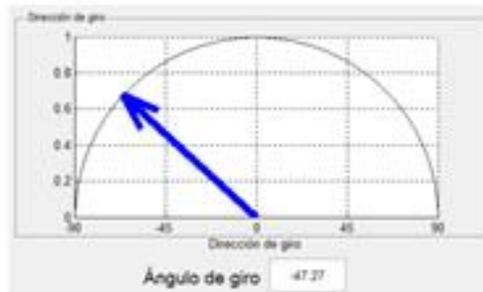
Objeto en movimiento  
[bebé con andadora]

(a) Imagen RGB

(b) Imagen de profundidad



(c) Representación gráfica del sistema difuso

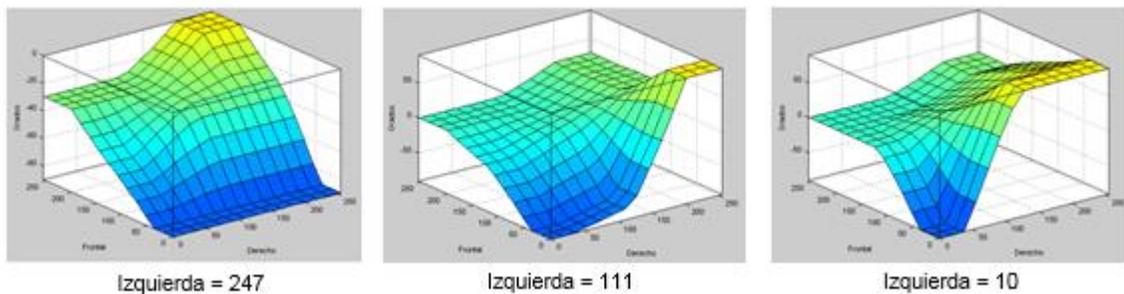


(d) Gráfica de dirección de giro

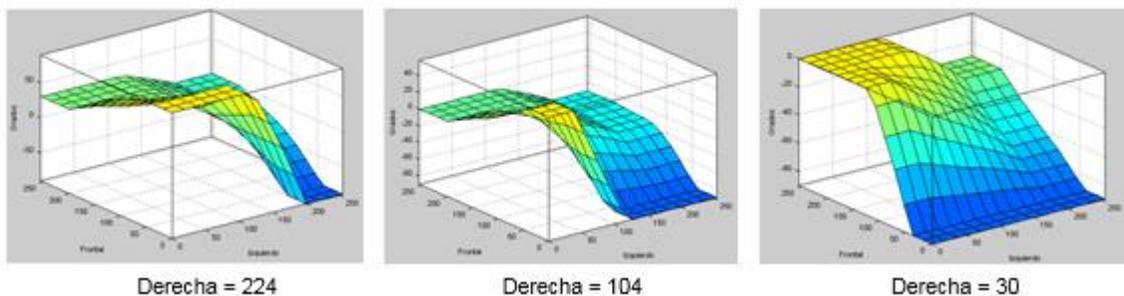
**Figura 6.** .Comportamiento del sistema en ambientes domésticos, con ligeros movimientos

El comportamiento de cada variable en función de los datos de profundidad puede ser visualizado en un gráfico 3D. Donde el eje z es el ángulo de giro de la plataforma móvil, el eje x y y son los valores de profundidad promedio acorde a dos variables de entrada. De esta forma es posible examinar el comportamiento de la variable de salida (ángulo de giro) utilizando combinaciones entre dos entradas que varían como producto de cambios en el entorno (obstáculos o ruido) y una entrada constante, generando los siguientes escenarios:

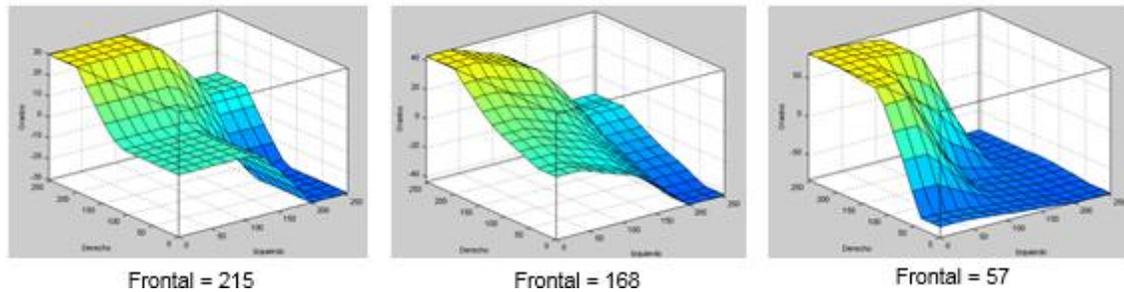
- Comportamiento con cambios en las sub-imágenes de la parte frontal y el lado derecho, y sin cambios al lado izquierdo (véase Figura 7).
- Comportamiento con cambios en las sub-imágenes de la parte frontal y el lado izquierdo, y sin cambios al lado derecho (véase Figura 8).
- Comportamiento con cambios en las sub-imágenes del lado izquierdo y el lado derecho, y sin cambios en la parte frontal (véase Figura 9).



**Figura 7.** Representación de la relación entre las variables de entrada Frontal y Derecha con respecto a la variable de salida Ángulo.



**Figura 8.** Representación de la relación entre las variables de entrada Frontal e Izquierda con respecto a la variable de salida Ángulo



**Figura 9.** Representación de la relación entre las variables de entrada Derecha e Izquierda con respecto a la variable de salida Ángulo

## 5. Conclusiones

En este trabajo se presentó un sistema de navegación reactiva difusa para generar giros suaves y anticiparse a futuras colisiones utilizando imágenes de profundidad adquiridas mediante el Kinect. El sistema propuesto cuenta con un conjunto de 27 reglas predictivas que buscan evitar: colisiones futuras considerando las profundidades más alejadas, colisiones con objetos muy cercanos u objetos imprevistos, y giros pronunciados que puedan poner en riesgo la integridad de la plataforma móvil.

La reconstrucción morfológica aplicada permitió reducir la cantidad de información desconocida en la imagen de profundidad, y obtener una mayor definición de los objetos inmersos en la escena. El proceso de reducción de bits por pixel realizado sobre la imagen de profundidad original llevó a reducir el costo computacional de 16 bits a 8 bits, haciendo más factible su implementación en hardware reconfigurable o sistemas basados en microcontroladores. La elección de un sistema de inferencia Sugeno también contribuye a reducir el procesamiento de información.

Las pruebas realizadas corroboraron la viabilidad del sistema para la obtención de giros suaves de plataformas móviles. El sistema de navegación reactiva difusa resulta conveniente en aplicaciones que requieran tolerancia a la incertidumbre de información y/o mecanismos de seguridad ante posibles colisiones.

Una aplicación propuesta para el sistema de navegación reactiva difusa es su incorporación como elemento de control de una silla de ruedas robotizada, considerando que en estos dispositivos el cuidado de la integridad de usuario es la prioridad, siendo la reacción ante objetos imprevistos y la navegación mediante giros suaves dos elementos de utilidad. La integración de algoritmos para discriminar los objetos inmersos en la escena o el suelo como tal, es una línea de trabajo a seguir.

## Referencias

Acosta, G., Gallardo, J., & Pérez, R. (2016). Arquitectura de control reactiva para la navegación autónoma de robots móviles. *Ingeniare*, 24 (1), 173-181.

Aguirre, E., Gómez, M., Muñoz, R., & Ruiz, C. (2003). Un sistema multi-agente que emplea visión activa y ultrasonidos aplicado a navegación con comportamientos difusos. En *IV Workshop en Agentes Físicos* (pp. 63-74). Alicante, España: Universidad de Alicante.

Amanda, W. (2010). *Programming Mobile Robots with Aria and Player*. London: Springer

Andersen, M.R., Jensen, T., Lisouski, P., Mortensen, A.K., Hansen, M.K., Gregersen T., & Ahrendt, P. (2012). Kinect Depth Sensor Evaluation for Computer Vision Applications (Technical report ECE-TR-6, Aarhus University, Denmark). Recuperado de [http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske\\_rapporter/Technical\\_Report\\_ECE-TR-6-samlet.pdf](http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske_rapporter/Technical_Report_ECE-TR-6-samlet.pdf)

Benavidez, P. & Jamshidi M. (2011). Mobile robot navigation and target tracking system. In *6th International Conference on System of Systems Engineering*, (pp. 299-304). Albuquerque, New Mexico, USA: IEEE Reliability Society & IEEE Systems, Man, and Cybernetics Society. doi: 10.1109/SYSOSE.2011.5966614

Bueno, M. & Rios, L. H. (2008). Implementación de comportamientos para navegación inteligente de robots móviles. *Tecnura*, 11 (22), 40-52.

Chica, M. T., Pineda, A. L., & Esmeral, M. A. (2009). Sistema de navegación para robots móviles utilizando fusión sensorial. *Tecnura*, 13 (25), 128-135.

Cruz, L., Lucio, D., & Velho, L. (2012). Kinect and RGBD Images: Challenges and Applications. In *25th Conference on Graphics, Patterns and Images Tutorials*,

(pp. 36–49). Ouro Preto, Brazil: IEEE Computer Society & Brazilian Computer Society. doi: 10.1109/SIBGRAPI-T.2012.13

Csaba, G., (2013). Improvement of an adaptive fuzzy-based obstacle avoidance algorithm using virtual and real kinect sensors. In 9th International Conference on Computational Cybernetics (pp. 113–120).Tihany, Hungary: IEEE Hungary Section & IEEE Systems, Man, and Cybernetics Society. doi: 10.1109/ICCCyb.2013.6617572

Csaba, G. & Vámosy Z. (2012). Fuzzy based obstacle avoidance for mobil robots with Kinect sensor. In 4th IEEE International Symposium on Logistics and Industrial Informatics (pp. 135–144). Smolenice, Eslovaquia: IEEE Hungary Section. doi: 10.1109/LINDI.2012.631947

Díaz, C. A. & Romero, C. A. (2012). Navegación de robot móvil usando Kinect, OpenCV y Arduino. *Prospectiva*, 10 (1), 71–78. doi: 10.15665/rp.v10i1.398

El-laithy, R. A., Huang, J.; & Yeh, M. (2012). Study on the use of Microsoft Kinect for robotics applications. In Position Location and Navigation Symposium (pp. 1280–1288). Myrtle Beach, South Carolina, USA: IEEE Aerospace and Electronic Systems Society & Institute of Navigation. doi: 10.1109/PLANS.2012.6236985

Jang, J-S.R, Sun, C-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. New Jersey, USA: Prentice Hall.

Kelly, A. (2013). *Mobile Robotics: Mathematics, Models, and Methods*. New York, USA: Cambridge University Press.

Kornuta, C. & Marinelli, M. (2013). Estudio comparativo de la eficiencia entre controladores difusos del tipo Mandani y Sugeno. En XV Workshop de Investigadores en Ciencias de la Computación. (pp. 926–929). Paraná, Argentina: Universidad Autónoma de Entre Ríos.

Kuno, Y., Shimada N., & Shirai, Y. (2003). A Robotic Wheelchair Based on the Integration of Human and Environmental Observations. *IEEE Robotics & Automation Magazine*, 10 (1), 26-34.

Mester, G., (2010). Intelligent mobile robot motion control in unstructured environments. *Acta Polytechnica Hungarica*, 7 (4), 153–165.

Mittal, R., Goyal, D. (2014). Autonomous Navigation of Smart Wheelchair using Kinect Camera. *International Journal of Engineering and Technical Research*. 2 (4), 357-358.

Mohan, V. (2005). Fuzzy logic controller for an autonomous mobile robot (Master's thesis, Cleveland State University, Ohio). Recuperado de <http://www.eng.auburn.edu/~tropical/courses/5530%202011C%20Robots%20Fall%2011/literature/Mohan%20Thesis%20fuzzy%20logic%20wall%20follow.pdf>

Murphy, R. R. (2000). An introduction to AI robotics. Cambridge, Massachusetts, USA: The MIT Press.

Negnevitsky, M. (2005). Artificial Intelligence: A Guide to Intelligent Systems (2nd ed.). England: Addison-Wesley.

Ollero A. (2001). Robótica: manipuladores y robots móviles. Barcelona, España: Marcombo Alfaomega.

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9 (1), 62–66. doi:10.1109/TSMC.1979.4310076

Parker, J. (2010). Algorithms for image processing and computer vision. 2a Ed. USA: Wiley Publishing, Inc.

Parra, H., Ríos L. H., & Bueno M. (2007). Navegación de robots móviles mediante comportamientos utilizando lógica difusa. Scientia Et Technica, 13 (34), 79–83.

Ramírez, J. M., Gómez, P., Martínez, J., & López, F. (2011). A LabVIEW-based autonomous vehicle navigation system using robot vision and fuzzy control. Ingeniería Investigación y Tecnología, 12(2), 129–136.

Suárez, A. F. & Loaiza, H. (2015). Implementación de un esquema de navegación reactiva con sensores RGB-D. UIS Ingenierías, 14 (1), 7-19.

Tzafestas, S. G. (2014). Introduction to Mobile Robot Control. Elsevier

Videre Design (2009). ERA Mobile Robot User's Manual (Rev H, October 2009). Videre Design LLC.

Villaseñor, U. G., González, M. A., Sotomayor, A., Gorrostieta, E., Pedraza, J. C., Vargas, J. E. & Tovar S. (2010). Desarrollo de un sistema de navegación para robots móviles mediante diferentes patrones de comportamientos. En VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico (pp. 407-412). Cuernavaca, Morelos, México: EEE sección Morelos y AMIME sede Morelos.

Vincent, L. (1993). Morphological gray scale reconstruction in image analysis: applications and efficient algorithms. IEEE Transactions on Image Processing, 2 (2), 176–201.

Webb, J.& Ashley, J. (2012).Beginning Kinect Programming with the Microsoft Kinect SDK. New York, USA: Apress.

Yerubandi, V., Reddy, Y. M., & Kumar, M. V. (2015).Navigation system for an autonomous robot using fuzzy logic. International Journal of Scientific and Research Publications.5 (2). Recuperado de <http://www.ijsrp.org/research-paper-0215/ijsrp-p3807.pdf>

## Notas biográficas:



**Claudia Cruz Martínez** nació en Oaxaca, México en 1989. Recibió el grado de Ingeniero en Computación por la Universidad del Istmo en 2014. Fue becaria del Cuerpo Académico de Ingeniería en Computación. Actualmente se encuentra estudiando

la Maestría en Ciencias Computacionales en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Su área de interés se centra en el uso de la visión 3D, cómputo de alto desempeño y aplicaciones de tiempo real enfocados al SLAM visual.



**Ignacio Algreto Badillo** . recibió los grados de Ingeniero en Electrónica y de Maestro y Doctor en Ciencias Computacionales por parte del Instituto Tecnológico de Puebla (ITP) y del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE),

respectivamente. Él se desempeña como Profesor de Tiempo Completo en la Universidad Politécnica de Tlaxcala. Él es miembro del SNI desde el 2011, tiene más de 70 artículos científicos, es revisor en varios comités nacionales e internacionales, lidera varios proyectos y sus líneas de investigación son enfocadas en el diseño y desarrollo de sistemas digitales, arquitecturas reconfigurables, plataformas basadas en el concepto radio software, seguridad, prototipos didácticos, sistemas criptográficos, implementaciones en FPGA,

sistemas basados en microcontroladores y microprocesadores y aceleración en hardware para aplicaciones específicas.



**J. Jesús Arellano Pimentel** es ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Morelia en 2002 y Maestro en Ciencias en Ingeniería Eléctrica con opción en Sistemas Computacionales por la Universidad Michoacana de San Nicolás de Hidalgo en 2005. Actualmente se desempeña como Profesor-Investigador de Tiempo Completo adscrito a la carrera de Ingeniería en Computación en la Universidad del Istmo campus Tehuantepec, Oaxaca, México. También es miembro activo del Cuerpo Académico de Ingeniería en Computación de la UNISTMO. Además ha obtenido el Reconocimiento a Perfil Deseable PRODEP durante los periodos 2013-2016 y 2016-2019. Sus áreas de interés incluyen: software educativo, realidad virtual, prototipos didácticos, compiladores, y robótica móvil.



**Francisco Aguilar Acevedo** es ingeniero en electrónica por la Universidad Tecnológica de la Mixteca. Tiene una maestría en ingeniería mecatrónica por el Centro Nacional de Investigación y Desarrollo Tecnológico. Es Profesor-Investigador de Tiempo Completo en la carrera de Ingeniería en Computación en la Universidad del Istmo Campus Tehuantepec, Oaxaca, México. También es miembro activo del Cuerpo Académico de Ingeniería en Computación de la UNISTMO. Sus áreas de interés son la dirección de proyectos de ingeniería, los sistemas embebidos, la instrumentación electrónica y la automatización de procesos.



**Ernesto Cortés Pérez** recibió el grado de M. C. en Ciencias de la Computación por ITA-LITI (Laboratorio de Investigación en Tecnologías Inteligentes) en el Instituto Tecnológico de Apizaco, Tlaxcala, México. Desde 2007 ha sido Profesor-Investigador de Tiempo Completo en la Universidad del Istmo UNISTMO Campus Tehuantepec, Oaxaca, México. También es miembro activo del Cuerpo Académico de Ingeniería en Computación de la UNISTMO. Su actual área de interés incluye: sistemas inteligentes, lógica difusa, clasificación de patrones, redes neuronales artificiales, algoritmos bio-inspirados y visión artificial.



**Luis Alberto Morales Rosales** recibió el grado de ingeniero en Sistemas Computacionales del Instituto Tecnológico de Colima. Los grados de maestría y doctorado del Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE). Ha sido Profesor-Investigador de Tiempo Completo en la Universidad Politécnica de Puebla (2007-2010). Coordinó y fundó el programa de posgrado en sistemas computacionales del Instituto Tecnológico Superior de Misantla (2010-2016), cuenta con reconocimiento al perfil deseable por parte de PRODEP desde 2011. Actualmente es catedrático CONACYT asignado a la Universidad Michoacana de San Nicolás de Hidalgo. Sus líneas de investigación son enfocadas al cómputo móvil, sistemas distribuidos, desarrollo de algoritmos bio-inspirados, computación inteligente, y sistemas de transporte inteligente.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

# Los Sistemas Gestores de Flujos de Trabajo en la Gestión de Procesos Software

## The Workflow Management Systems for Software Process Management

Víctor Hugo Menéndez Domínguez<sup>1</sup>  
mdoming@correo.uady.mx

María Enriqueta Castellanos Bolaños<sup>1</sup>  
enriqueta.c@correo.uady.mx

<sup>1</sup> Universidad Autónoma de Yucatán, México

**Resumen:** Hoy en día, las empresas de software se ven obligadas a mejorar su rendimiento para poder sobrevivir en un entorno altamente competitivo: los clientes demandan una mayor calidad, menor plazo de entrega, mejor servicio, atención personalizada, etc. En este trabajo se describen a los procesos software así como los flujos de trabajo. Su combinación y automatización mejora el flujo de información asociado a las actividades de dichos procesos, lo que facilita el control y permite mejorar el rendimiento, reducir costos y plazos, minimizar errores y garantiza una mayor calidad, entre otras ventajas.

**Palabras clave:** Proceso software, flujo de trabajo, Sistemas Gestores de Flujos de Trabajo, Workflow Management System, WfMS

**Abstract:** Nowadays, software companies are forced to improve their performance in order to survive in a highly competitive environment: customers

demand higher quality, shorter delivery, best service, personalized attention, etc. This paper describes the software processes and workflows. Their combination and automatization improve the flow of information associated with the activities of such processes, which facilitates control and improves performance, reduces costs and time, minimizes errors and ensures greater quality, among other benefits.

**Keywords:** Software process, workflows, WorkFlow Management System, WfMS

# 1. Introducción

Un proceso software se define como un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad (Pressman, 2014). Es decir, un proceso software es un conjunto de actividades y resultados asociados que producen un producto software (Sommerville, 2015).

El proceso software constituye la base para el control de la gestión de proyectos de software y establece el contexto en el cual se aplican los métodos, técnicas, se generan los productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.) se establecen los fundamentos, se asegura la calidad, y el cambio se maneja de manera apropiada (Pressman, 2014; Berander y Wohlin, 2003).

Distintos procesos software organizan las actividades de diferentes formas y las describen con diferente nivel de detalle. El tiempo de cada actividad varía, así como los resultados. Aunque existen muchos procesos software diferentes, existen actividades fundamentales que son comunes para todos ellos. Los detalles (tareas) serán muy diferentes en cada caso, pero las actividades son iguales. La adaptación del proceso software es esencial para el éxito del proyecto. (Chan y Leung, 1997)

Para Sommerville (2015), estas actividades fundamentales son cuatro:

1. Especificación del software: debe definirse la funcionalidad del software y las restricciones sobre su operación.

2. Desarrollo del software: debe producirse software que cumpla con la especificación.
3. Validación del software: el software debe evaluarse para asegurar qué es lo que el cliente requiere.
4. Evolución del software: el software debe evolucionar para cumplir con los cambios requeridos por el cliente.

Por otro lado, Pressman (2014) establece estas actividades de forma distinta:

1. Comunicación: implica una intensa colaboración y comunicación con los clientes (stakeholders) además abarca la investigación de requisitos y otras actividades relacionadas.
2. Planeación: establece un plan para el trabajo de la ingeniería de software. Describe las tareas técnicas que deben realizarse, los riesgos probables, los recursos que serán requeridos, los productos del trabajo que han de producirse y un programa de trabajo.
3. Modelado: abarca la creación de modelos que permiten al desarrollador y al cliente entender mejor los requisitos del software y el diseño que logrará satisfacerlos.
4. Construcción: combina la generación del código (ya sea manual o automatizado) y la realización de pruebas necesarias para descubrir errores en el código.
5. Despliegue: el software (como entidad completa o un incremento completado de manera parcial) que se entrega al cliente, quién evalúa el producto recibido y proporciona información basada en su evaluación.

Además, define una serie de actividades paralelas que son aplicables a lo largo de todos los procesos software y se enfocan de modo principal en la gestión, el rastreo y el control del proyecto (Pressman, 2014):

1. Seguimiento y control del proyecto de software: permite que el equipo de software evalúe el progreso comparándolo con el plan de proyecto y así tomar las acciones necesarias para mantener el programa.
2. Gestión del riesgo: evalúa los riesgos que pudieran afectar los resultados del proyecto o la calidad del producto.
3. Aseguramiento de la calidad del software: define y conduce las actividades requeridas para asegurar la calidad del software.
4. Revisiones técnicas formales: evalúa los productos del trabajo de la ingeniería de software en un esfuerzo encaminado a descubrir y eliminar los errores antes de que estos se propaguen hacia la siguiente acción o actividad.

5. Medición: Define y recolecta mediciones del proceso, el proyecto y el producto para ayudar al equipo a entregar software que satisfaga las necesidades del cliente se puede usar en conjunto con todas las otras actividades del marco de trabajo o actividades sombrija.
6. Gestión de la configuración del software: maneja los efectos del cambio a través del proceso del software.
7. Gestión de la reutilización: define los criterios para la reutilización de productos de trabajo (se incluyen componentes de software) y establece mecanismos para la creación de componentes reutilizables.
8. Preparación y producción del producto de trabajo: abarca las actividades requeridas para crear productos del trabajo como modelos, documentos, registros, formatos y listas.

Todos estos procesos software son descritos mediante modelos que se presentan desde una perspectiva particular. Por su naturaleza, los modelos son simplificaciones, por lo tanto un modelo de procesos software es una abstracción de un proceso real. Algunos ejemplos de estos tipos de modelos que se pueden producir son (Sommerville, 2015):

- Modelo de flujo de trabajo: muestra la secuencia de actividades en el proceso en conjunción con sus entradas, salidas y dependencias.
- Modelo de flujo de datos o de actividad: representa el proceso como un conjunto de actividades, cada una de las cuales lleva alguna transformación en los datos. Muestra cómo la entrada en el proceso, se transforma en una salida.
- Modelo de rol/acción: representan los roles de la gente involucrada en el proceso de software y las actividades de las que son responsables.

Los modelos genéricos de procesos deben adaptarse para que los utilice un equipo de proyecto de software. Para lograrlo se han desarrollado herramientas destinadas a ayudar a las organizaciones de software a analizar sus procesos actuales, organizar sus tareas en flujos de trabajo, controlar y monitorear su progreso, y administrar su calidad técnica.

Su origen está en la necesidad de mantener la integridad en los documentos generados como parte de un proceso en el que está involucrado un grupo de personas a lo largo de un período de tiempo.

## 2. Flujos de Trabajo

Un flujo de trabajo (o Workflow) es la automatización de un proceso de negocio (o de software), total o parcial, en la cual documentos, información o tareas son pasados de un participante a otro para efectos de su procesamiento, de acuerdo a un conjunto de reglas establecidas para obtener, o contribuir a los objetivos de la empresa (Hollingsworth, 1995).

Un Workflow involucra un número de pasos lógicos, cada uno conocido como una actividad. Una actividad puede involucrar interacción manual con un usuario o con otro flujo de trabajo o la actividad puede ser ejecutada utilizando algún recurso-máquina (Allen, 2001).

En otras palabras, un flujo de trabajo es, según Plesums (2002):

- Una secuencia de tareas donde intervienen participantes,
- Cada participante desempeña un papel (rol),
- Unos productos se manipulan y se transfieren entre participantes,
- Se hace uso de una información que fluye entre las tareas,
- De acuerdo con un conjunto de reglas y procedimientos,
- Para alcanzar un objetivo.

Los flujos de trabajo son un camino para la información, para reducir tiempo, dinero y esfuerzo en la ejecución de un proceso de negocio.

Mientras que los Workflow pueden ser gestionados manualmente, en la práctica están organizados en el contexto de un sistema informático que proporciona un soporte computarizado para una sistematización procedural (Hollingsworth, 1995; Pressman, 2014).

Dicha sistematización permite mejorar el rendimiento de los procesos, reducir los costos y los plazos, minimizar errores y garantizar una mayor calidad. Los

sistemas que permiten automatizar los procesos de negocio reciben el nombre de Sistemas Gestores de Flujos de Trabajo y se apoyan en la utilización de las Tecnologías de la información.

### 3. Sistemas Gestores de Flujos de Trabajo

La gestión de flujos de trabajos para procesos requiere de herramientas software que incluyan elementos para definirlos, ejecutarlos, interfaces para utilizarlos, compartirlos, monitorearlos y controlarlos (Pressman, 2014).

Esta sección describe estas herramientas software llamadas Sistemas Gestores de Flujos de Trabajo (Workflow Management System, WfMS) y su relación con la gestión de procesos software.

La necesidad de reorganizar y mejorar los procesos de negocios y los avances en la tecnología de la información han creado un gran mercado para los Sistemas Gestores de Flujos de Trabajo (Casati, et al., 1996).

Un Sistema Gestor de Flujos de Trabajo (o simplemente sistema Workflow) proporciona una sistematización de los procesos de negocios mediante la gestión de la secuencia de actividades que los componen asignándolas a personas o procesos informáticos que las llevarán a cabo (Hollingsworth, 1995).

Los sistema Workflow definen, crean y gestionan la ejecución de flujos de trabajo mediante el uso de software (motores de ejecución) que son capaces de interpretar la definición del proceso, interactuar con los participantes y cuando es requerido ejecutar herramientas y aplicaciones (Berger et al., 1997).

Los Sistemas Gestores de Flujos de Trabajo ayudan a manejar los procesos de negocios asegurando que las tareas son priorizadas para ser llevadas a cabo (Plesums, 2002):

- Tan pronto como sea posible,
- Por la persona correcta,

- En el orden correcto.

Los sistemas Workflow se originaron como una forma de reducir el tiempo y costo de llevar a cabo los procesos de negocios y asegurar que las tareas sean llevadas a cabo consistentemente para mejorar la calidad. Se basan sobre la idea de que algunas cosas son realizadas más efectivamente por las computadoras que por las personas: búsqueda, control, etc. (Halliday, 2001)

Generalmente son utilizados por las empresas cuando es necesario sistematizar procesos de negocios que tienden a ser repetitivos, bien estructurados y con gran manejo de datos, o bien cuando se requiere mantener el control y la integridad sobre las modificaciones realizadas en un documento por un grupo de personas con el objetivo de lograr una meta común. Esto está relacionado con procesos administrativos relacionados con reportes de venta, ordenes de compras, control de inventarios, etc.

### 3.1 Características

Los Sistemas Gestores de Flujos de Trabajo permiten, diseñar, ejecutar y monitorear los flujos de trabajos en entornos distribuidos e incluso heterogéneos, permitiendo una gestión y ejecución eficiente de los procesos (Casati et al., 1996).

Una característica básica que debe proporcionar un sistema Workflow es la flexibilidad. En un entorno tan cambiante, las compañías necesitan refinar sus procesos constantemente con el propósito de cubrir las necesidades y oportunidades propuestas por nuevas tecnologías, nuevos requerimientos de mercado y nuevas leyes (Casati et al., 1996).

Las funciones más comunes que proporcionan los Sistemas Gestores de Flujos de Trabajo son (Plesums, 2002):

- Asignación de tareas al personal,

- Aviso al personal de tareas pendientes,
- Permitir la colaboración en las tareas comunes,
- Optimización de recursos humanos y técnicos, alineándolos a la estrategia de la empresa,
- Automatización de las secuencias de los procesos de negocio y optimización de las mismas,
- Agilización de los procesos de negocio y como resultado un mejor servicio al cliente,
- Control y seguimiento de dichos procesos,
- Interfaz para el acceso a sistemas de datos,
- Registro y monitoreo de acciones realizadas.

## 3.2 Ventajas

Los beneficios que se obtienen al emplear un Sistema Gestor de Flujos de Trabajo pueden ser divididos en tangibles e intangibles (Plesums, 2002; Prior, 2003):

Beneficios tangibles:

- Reducción en los costos de operación,
- Enfoque a la disminución de trámites,
- Mejora en la productividad,
- Disminución en los tiempos de procesamiento,
- Reducción en los tiempos de respuesta,
- Monitoreo de estado, recursos disponibles,
- Asignación de tareas de forma adecuada,
- Control de procesos,
- Integración con otras aplicaciones.

Beneficios intangibles:

- Mejora en los servicios,
- Mejora en las condiciones de los empleados,
- Mejora en la gestión del cambio,
- Calidad,
- Mejora en la comunicación,
- Soporte para la toma de decisiones,
- Mejora en la capacidad de planeación,
- Mejora en la capacidad de distribución,
- Comunicación entre organizaciones.

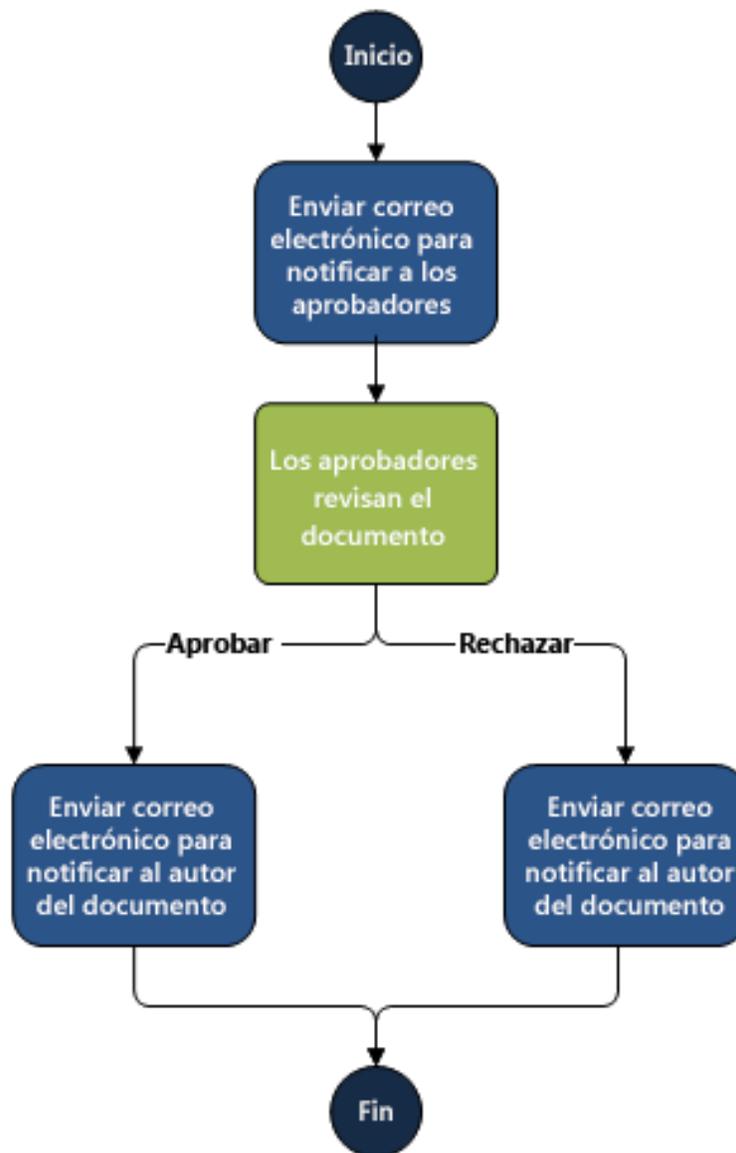
### 3.3 Requerimientos

Emplear un sistema Workflow para resolver una problemática en una empresa implica primeramente establecer de forma clara un modelo de procesos de negocios que incorpore los elementos que serán incluidos en el sistema:

- Tareas
- Personas (Usuarios),
- Roles,
- Reglas de transición,
- Datos,
- Eventos,
- Plazos (Deadlines),
- Procesos,
- Políticas.

Generalmente esta modelación se realiza utilizando un lenguaje de modelado de procesos como BPMN (OMG, 2006), el cual proporciona una notación que es fácilmente entendible para todos los usuarios relacionados con el sistema (usuarios finales, analistas, desarrolladores y gestores de procesos). Su finalidad

es proporcionar un puente entre el diseño del proceso de negocios y su implementación dentro de un sistema Workflow.



**Figura 1.** Ejemplo de un modelo de proceso de negocio (Microsoft, 2016).

Posteriormente está el emplear o desarrollar un sistema Workflow que se adapte a las necesidades de la empresa. En este sentido existen varias opciones en el mercado.

Por ejemplo, Microsoft Office SharePoint Server 2007 (Microsoft, 2016) proporciona herramientas que facilitan la colaboración en la edición de documentos y administrar tareas del proyecto tales como ver la lista de flujos de trabajo disponibles para un documento, iniciar un flujo de trabajo, reasignar una tarea del flujo de trabajo o completarla. Ofrece Windows Workflow Foundation, una API que permite al programador crear flujos de trabajo con código personalizado que pueden utilizarse dentro del mismo entorno e integrarse con otros sistemas.

Por otra parte, Flokzu (<http://www.flokzu.com/ES>), intalio (<http://www.intalio.com/>) o jBPM (<http://www.jbpm.org/>) son Sistemas Gestores de Flujos de Trabajo que proporcionan una serie de características que las hacen una opción interesante:

- Trabajo en equipo,
- Crear modelos de procesos en formato BPMN,
- Diseñar formularios de captura,
- Almacenamiento en la nube,
- Integración con otros sistemas,
- Generación de estadísticas, entre otros.

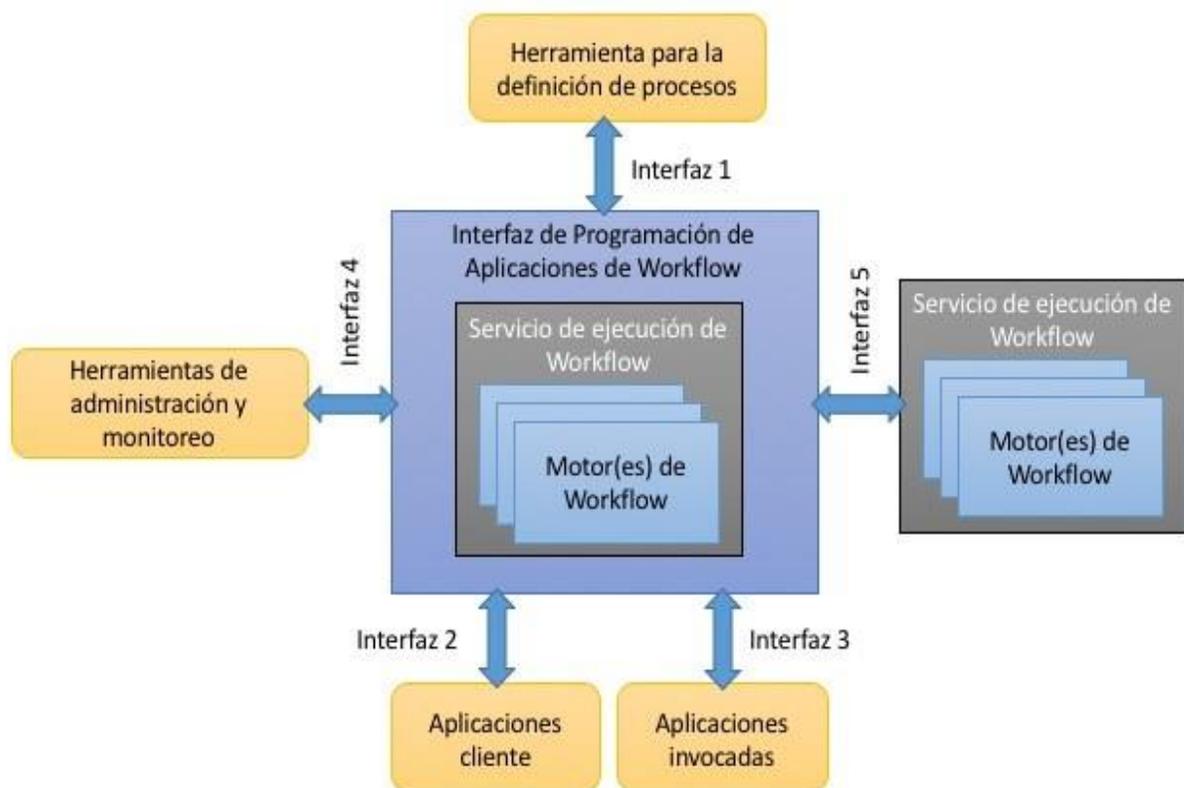
### 3.4 Modelo de Referencia

Con el propósito de establecer un estándar para el desarrollo de Sistemas Gestores de Flujos de Trabajo, la Workflow Management Coalition (WfMC) ha establecido un modelo de referencia para Sistemas Gestores de Flujos de Trabajo (Hollingsworth, 1995) mostrado en la figura 2.

En el modelo adoptado hay una separación entre los procesos y el control de la lógica de las actividades. Esta separación permite la integración de las diversas herramientas con una aplicación particular. La interacción del servicio de

ejecución con los recursos externos se da por una de las dos interfaces siguientes:

1. La interfaz de las Aplicaciones cliente, a través de la cual el Motor de Workflow interactúa con el manejador de la lista de trabajos (Worklist), responsable de organizar el trabajo por intermedio de un recurso de usuario. Es responsabilidad de este manejador hacer progresar cada elemento de la Worklist.
2. La interfaz de las Aplicaciones invocadas, la cual le permite al motor de Workflow activar una herramienta para realizar una actividad particular. Esta interfaz podría ser basada en un servidor, es decir no existe la interacción con el usuario.
3. Milk



**Figura 2.** Modelo de Referencia WfMS.

A continuación se describen los principales componentes del modelo:

**Motor de Workflow (Workflow Engine).** El motor de Workflow es el software que provee el control del ambiente de ejecución de una instancia de flujo de trabajo. Típicamente el motor provee facilidades para:

- • Interpretación de la definición de procesos.
- • Control de las instancias de los procesos: creación, activación, terminación, etc.
- • Navegación entre actividades.
- • Soporte de interacción con el usuario.
- • Control de datos al usuario o hacia aplicaciones.
- • Invocación de aplicaciones externas.

**Servicio de ejecución de Workflow (Workflow Enactment Service).** Este componente interpreta la descripción de procesos y controla las diferentes instancias de los procesos, secuencia de actividades, adiciona elementos a la lista de trabajo de los usuarios, e invoca aplicaciones necesarias. Todas estas tareas son hechas por uno o más motores de Workflow, los cuales manejan la ejecución de las distintas instancias de varios procesos.

**Interfaz de Programación de Aplicaciones de Workflow (WAPI).** Las WAPI pueden ser vistas como un conjunto de API's (Application Programming Interface) y funciones de intercambio soportadas por el servicio de representación de Workflow. Las WAPI permiten la interacción del servicio de representación de Workflow con otros recursos y aplicaciones.

**Herramientas de definición de procesos (interfaz 1).** Existe una gran variedad de herramientas utilizadas para el análisis de procesos. Estas herramientas pueden variar desde las más informales hasta las más formales y sofisticadas. La salida de este proceso de modelado y diseño es una "definición de procesos" la cual puede ser interpretada en tiempo de ejecución por el o los motores de Workflow.

**Aplicaciones clientes (interfaz 2).** En el modelo planteado, la interacción entre las aplicaciones clientes y el motor de Workflow está sostenido en gran parte por el concepto de lista de trabajo ya descrito anteriormente. Parte de la información almacenada en la lista de trabajo es utilizada para transmitirle al manejador de la lista de trabajo qué aplicaciones hay que invocar. La interfaz entre una aplicación cliente de Workflow y el motor de Workflow debe ser lo suficientemente flexible en los siguientes puntos: identificadores de procesos y actividades, estructuras de datos, diferentes alternativas de comunicación.

**Aplicaciones invocadas (interfaz 3).** Esta interfaz está orientada a interactuar con agentes de una aplicación, o con toda la aplicación. Dichas aplicaciones deben estar orientadas al contexto general de un sistema de Workflow, es decir, deben poder interactuar directamente con el motor de Workflow. La aplicación invocada es manejada localmente por un motor de Workflow, usando la información suministrada en la definición del proceso para identificar la naturaleza de la actividad. La aplicación invocada puede ser local al motor de Workflow, es decir, residente en la misma plataforma, o estar en otra plataforma dentro de una red. En este caso la definición del proceso debe contener información necesaria para poder encontrar la aplicación que se va a invocar.

**Funciones de interoperabilidad WAPI (interfaz 4).** Existen dos aspectos necesarios para la interoperabilidad:

- Alcance de la interpretación común de la definición de procesos que será realizada.
- Soporte en tiempo de ejecución para el intercambio de diferentes tipos de información de control y transferencia de los datos relevantes del Workflow, y/o de las aplicaciones entre los distintos servicios de representación.

**Herramientas de administración y monitoreo (interfaz 5).** El propósito de esta interfaz es permitir una vista completa del estado del flujo de trabajo, además de poder realizar auditorías sobre los datos del sistema.

## 3.5 Clasificación

Debido a la diversidad de procesos de negocio que existen dentro de las empresas, es posible clasificar a los Sistemas Gestores de Flujos de Trabajo según su funcionalidad principal.

Para Plesums (2002), estos pueden ser clasificados en:

- **Producción.** Frecuentemente este tipo de sistemas Workflow es llamado Workflow de Transacciones. Esto se debe a que la transacción en una base de datos es considerada la clave de todo proceso. Este tipo de sistemas Workflow es el segmento más grande en el mercado. En general automatizan procesos de negocios que tienden a ser repetitivos, bien estructurados y con gran manejo de datos.
- **Colaboración.** Los sistemas Workflow que resuelven procesos de negocios donde participa gente para lograr una meta común, son llamadas Workflow de Colaboración. Los Workflow de colaboración estructuran o semi-estructuran procesos de negocios donde participan personas, con el objetivo de lograr una meta común.
- **Administración.** El Workflow Administrativo es aquel que involucra procesos de administración en una empresa tales como órdenes de compra, reportes de ventas, etc. Estos Workflow's se emplean cuando existe una gran cantidad de procesos administrativos dentro de la empresa y es necesaria la distribución de soluciones a diferentes usuarios.

## 3.6 Gestión de Procesos Software

Dado el éxito que ha tenido los flujos de trabajo para la automatización y la reingeniería de procesos de negocios, ha atraído la atención de la ingeniería de software, donde se realizan esfuerzos para encontrar soluciones flexibles para la gestión de los procesos de desarrollo en sistemas grandes y complejos (Aversano, 2001).

Los procesos de desarrollo de software son un tipo especializado de procesos de negocios en el cual, documentos, información y tareas son pasados de un

participante a otro conforme a un conjunto de reglas conocidas como métodos de desarrollo (Barnes y Gray, 2000).

Los procesos software son complejos, y como todo proceso intelectual, se basa en el juicio humano. Debido a la necesidad de juzgar y crear, los intentos para automatizar estos procesos han tenido éxito limitado, esto debido a la gran diversidad de procesos software. No existe un proceso ideal y diferentes organizaciones ha desarrollado enfoques distintos para desarrollar software, explotando las capacidades del personal así como las características de las aplicaciones que se están desarrollando. Por lo tanto, aún dentro de una misma compañía pueden existir muchos procesos diferentes para desarrollar software (Sommerville, 2015).

El empleo de un paradigma de flujos de trabajo para la ejecución y gestión de los procesos de desarrollo de software ofrece una mayor apertura para interoperar con otros sistemas, crear entornos de ejecución distribuidos, facilitar el monitoreo y gestionar los recursos humanos (Chan y Leung, 1997).

De forma que los Sistemas Gestores de Flujos de Trabajo pueden ayudar a las organizaciones de software a analizar sus procesos actuales, organizar sus tareas, controlar y monitorear su progreso, y administrar su calidad técnica (Allen, 2001).

Estos sistemas permiten que una organización de software construya un modelo automatizado del marco de trabajo común del proceso, de los conjuntos de tareas y las actividades. El modelo puede analizarse para determinar el flujo de trabajo típico y examinar las estructuras del proceso alternativas que podrían conducir a la reducción del tiempo o costo del desarrollo (Sommerville, 2015).

Una vez creado un proceso aceptable es posible localizar, monitorear e incluso controlar todas las tareas de ingeniería del software definidas como una parte del modelo del proceso. Cada miembro del equipo de software puede emplear herramientas para la elaboración de una lista de verificación de las tareas de

trabajo que se deben desarrollar, los productos de trabajo que son imperativos en producir, y las actividades para el aseguramiento de la calidad que deben realizarse.

Los Sistemas Gestores de Flujos de Trabajo pueden coordinar el uso de otras herramientas de la ingeniería del software asistida por computadora que sean apropiadas para una tarea de trabajo particular. En algunos casos incorporan tareas de gestión del proyecto, como estimación, itinerario, rastreo y control (Aversano et al., 2001). Son más flexibles, y proporcionan a los gestores de proyectos de software cierto grado de libertad para decidir el nivel de detalle en la especificación de un proceso software (granularidad) para adoptar en la realización de un sistema y seleccionar de entre varios modelos de procesos (cascada, iterativos, espiral, etc.) para realizar distintos componentes del sistema.

Aunque los Sistemas Gestores de Flujos de Trabajo pueden ayudar a algunas actividades del proceso, no se prevé, al menos en los próximos años, de una automatización mayor en el diseño creativo del software realizado por los ingenieros relacionados con el proceso software (Sommerville, 2015).

Chan y Leung (1997) establecen que es necesario mantener algunas consideraciones con los sistemas Workflow actuales:

- Ejecución de procesos software de forma automática,
- Reusabilidad de actividades, estructuras, funcionamientos,
- Abstracción para modelar controles y datos,
- Expresividad para facilitar la modelación,
- Monitoreo del desempeño,
- Flexibilidad para realizar cambios de forma dinámica,
- Consistencia en los datos compartidos, manejo de transacciones,
- Verificación para detectar errores en la definición de procesos,
- Control de versiones,

- Integración de herramientas,
- Distribución de procesos entre el personal,
- Datos de estado para la gestión.

### 3.7 Esquema de Implementación

La implementación de un Sistema Gestor de Flujos de Trabajo en una empresa es similar a otros proyectos de tecnologías de información. Se debe establecer el ámbito, realizar un plan de implementación, desarrollar la tecnología, implementarla y evaluarla antes de ponerla en uso. Los principales pasos son (Allen, 2001):

1. **Definir el ámbito del proyecto.** Identificar los límites, iniciar en un departamento de misión no crítica. Identificar el tipo de sistema Workflow a ser implementado. Establecer recursos
2. **Planeación de la implementación.** Se presta atención a los detalles, se pretende entender claramente cómo los procesos de negocios trabajan, generalmente mediante entrevistas. Se identifica qué procesos pueden ser mejorados mediante el uso de los sistemas Workflow.
3. **Desarrollo de la aplicación.** Puede considerarse una parte complicada, pero la tecnología de hoy en día para Sistemas Workflow ha tenido numerosos avances, lo que ha permitido generar herramientas sencillas para crear definiciones de procesos y ejecutarlas en entornos. Además que existen API's que ayudan al desarrollador acceder a datos externos.
4. **Evaluación.** Esta actividad no es diferente a las de otros proyectos de implementación de software. Se define un plan de pruebas que cubre todas las eventualidades posibles. Es importante que se tenga un control en todo momento de la gestión de las reglas.
5. **Pre-implementación.** Es importante establecer un plan de capacitación para preparar al personal en el uso del sistema Workflow. Los cambios se dan no solo en la forma de realizar los procesos sino también en la concepción de los mismos.
6. **Post-implementación.** Una vez que la tecnología ha sido liberada es necesario monitorear la eficiencia de los nuevos métodos y aprender de la experiencia para que esta sea incrementada. La automatización crea eficiencia pero es necesario rediseñar los procesos para obtener los máximos beneficios de la combinación de estos elementos.

## 4. Lecciones Aprendidas

En los últimos años y a nivel mundial, los sistemas Workflow han generado gran interés por dos razones principales: una económica y otra tecnológica. La económica se basa en el reconocimiento que debido a la globalización de los mercados, el ambiente competitivo de negocios actual necesita la automatización de todas las actividades y procesos de trabajo, y no solo la automatización de algunas tareas individuales. La razón tecnológica resulta de la aparición de nuevos ambientes computacionales, capaces de integrar varias aplicaciones que antes operaban de manera independiente.

A pesar de que organizaciones de software utilizan procesos software diferentes para producir el mismo tipo de producto, es posible automatizar estos procesos. La automatización permite:

- Reducción drástica de tiempos de ciclo, por la agilización de las tareas administrativas y minimización de errores ,
- Mayor integración de las actividades del proceso ,
- Facilita la comunicación y coordinación entre las personas que realizan las tareas,
- Mejora el flujo de información asociado a las actividades del proceso,
- Facilita su control y medición de resultados.

Es decir la automatización mejora el rendimiento, reduce costos y plazos, minimiza errores y garantiza mayor calidad en el software desarrollado.

Los Sistemas Gestores de Flujos de Trabajo automatizan diferentes aspectos del flujo de la información: direccionan los trabajos en la secuencia correcta, proveen acceso a datos y documentos, y gestionan ciertos aspectos de la ejecución de un proceso. Mejoran la eficiencia y la estandarización de los procesos, contribuyendo a la reducción de costos dentro de una empresa, pero más importante aún es que es que permiten obtener un mayor conocimiento de los procesos, y así establecer estrategias para mejorar la calidad.

La gestión de los procesos software con estos sistemas puede monitorear el estado actual de las actividades y observar cómo evolucionan. Así, es posible detectar y solucionar situaciones que requieren tiempos no planificados (decisiones críticas). También permiten asignar tareas de forma automática mediante la definición de roles, así como establecer recursos de información que serán necesarios para poder realizarlas.

Los sistemas Workflow incorporan elementos para control de secuencias de tareas, quiénes realizan dicha secuencia, así mecanismos de control y monitoreo.

Los sistemas Workflow utilizan generalmente una de dos arquitecturas:

1. Formas electrónicas que proporciona una interfaz para tareas. Estas formas son transportadas por todas las fases de las tareas usando mensajes. Este tipo de productos permiten crear aplicaciones que son utilizadas para realizar un trabajo.
2. Aplicaciones externas que proporcionan interfaz para tareas. El entorno de proceso es controlado por un motor Workflow, el cual conserva un registro del progreso de cada instancia del proceso. Este tipo de productos proporcionan una gestión sofisticada del proceso.

## 5. Conclusión

Los Sistemas Gestores de Flujos de trabajo permiten el control de los procesos software y al mismo tiempo disminuyen la necesidad de intervenir para su gestión. Esto permite que haya menos retrasos en los tiempos destinados a la entrega de un producto o pérdidas como consecuencia de esos retrasos. Al mismo tiempo incrementa la calidad del servicio al responder de una forma más rápida y con el mejor personal disponible.

Además disminuye los costos de entrenamiento debido a que el trabajo puede ser guiado a través de complejos procedimientos, permite una disminución de los costos de gestión, incrementando el control y permitiendo a los

administradores concentrarse en el apoyo a los empleados y manejar casos especiales en vez de rutinarios.

Los sistemas Workflow requieren ciertas condiciones para que puedan ser implantados correctamente:

1. Los procesos deben tener tareas explícitas,
2. Reglas para determinar la lógica de transiciones entre tareas,
3. Las tareas utilizan recursos digitales de información,
4. Las tareas deben ser comunicadas a los trabajadores,
5. Hay la necesidad de un proceso de control.

Los elementos básicos de un Sistema de Workflow son

1. Herramientas de análisis, modelado y definición de procesos de negocio. Se representa la secuencia de actividades y de información de cada flujo (Diagramas de flujo de datos).a. Descripción de procesos mediante un lenguaje formal de definición. Incluye. Actividades elementales, operaciones asociadas, reglas de cada etapa y como es la transición entre etapas.
2. Motor del Sistema de Workflow: Interpreta las definiciones de los procesos y crea un entorno para su ejecución.a. Controla las etapas de ejecución de un procesob. Invoca a la persona o aplicación cada etapac. Distribuye las tareasd. Distintos medios de comunicación: Correo electrónico, mensajería, intercambio de datos, etc.
3. Interfaz del sistema de Workflow: con usuarios y aplicaciones. Para rellenar campos de un formulario o manipular un documento.a. Controla la interacción del motor con el usuario o aplicación transfiriendo los datos necesarios en cada casob. Gestiona las colas de trabajo (tareas pendientes, ...)

El modelo de referencia WfMC tiene varios propósitos para el desarrollo de nuevos sistemas Workflow:

1. Ofrece un vocabulario común para describir los procesos de negocios y varios aspectos de tecnologías de soporte que facilitan la automatización. Esto proporciona una base esencial para discusiones sobre cómo puede construirse un sistema Workflow.
2. Una descripción funcional de los componentes de software claves para un Sistema de Gestión de Flujos de Trabajo y cómo deben interactuar. Todo desarrollado con una tecnología neutra de manera que el modelo sea

independiente de cualquier arquitectura particular de producto así como tecnología de implementación.

3. La definición en términos funcionales (o abstractos), de la interfaz entre varios componentes de software claves debe facilitar el intercambio de información en una forma estandarizadas, esto es permitiendo el intercambio de información entre productos, cinco interfaces ha sido definidas.

Un sistema Workflow da a una empresa la posibilidad de automatizar sus procesos, reducir costos, y mejorar servicios. Organizaciones que no hayan evaluado esta tecnología podrían encontrarse en desventaja en un futuro.

## Referencias

Allen, R. (2001). Workflow: An Introduction. Workflow Handbook 2001. Recuperado de [http://www.wfmc.org/information/Workflow-An\\_Introduction.pdf](http://www.wfmc.org/information/Workflow-An_Introduction.pdf)

Aversano L., de Lucia A., Stefanucci, S. y Betti S. (2001). Introducing Workflow Management in Software Maintenance Processes. 17th IEEE International Conference on Software Maintenance (ICSM'01). Recuperado de <http://doi.ieeecomputersociety.org/10.1109/ICSM.2001.972757>

Barnes, A. y Gray, J. (2000). COTS, Workflow, and Software Process Management: An Exploration of Software Engineering Tool Development. Australian Software Engineering Conference. Recuperado de <http://doi.ieeecomputersociety.org/10.1109/ASWEC.2000.844579>

Berander P. y Wohlin, C. (2003). Identification of Key Factors in Software Process Management - A Case Study. International Symposium on Empirical Software Engineering (ISESE'03). Recuperado de <http://doi.ieeecomputersociety.org/10.1109/ISESE.2003.1237992>

Berger, M., Ellmer, E., Quirchmayr, G. y Zeitlinger A. (1997). Evaluating Workflow Management Systems. 8th International Workshop on Database and Expert Systems Applications (DEXA '97). Recuperado de <http://doi.ieeecomputersociety.org/10.1109/DEXA.1997.617325>

Casati, F., Ceri, S., Pernici, B., y Pozzi, G. (1996). Workflow Evolution. Proceedings of the 15th international Conference on Conceptual Modeling (Octubre 07 - 10, 1996). B. Thalheim, Ed. Lecture Notes In Computer Science, vol. 1157. Springer-Verlag, London, 438-455.

Chan, D. y Leung K. (1997). A Workflow Vista of the Software Process. 8th International Workshop on Database and Expert Systems Applications (DEXA '97). Recuperado de <http://doi.ieeecomputersociety.org/10.1109/DEXA.1997.617237>

Halliday, J., Shrivastava, K. y Wheeler, S. (2001). Flexible Workflow Management in the OPENflow System. Fifth IEEE International Enterprise Distributed Object Computing Conference. Recuperado de <http://doi.ieeecomputersociety.org/10.1109/EDOC.2001.950425>

Hollingsworth, D. WfMC. (1995). Workflow Reference Model, white paper. The Workflow Reference Model (WfMC-TC-1003). Recuperado de <http://www.wfmc.org/standards/docs/tc003v11.pdf>

Microsoft (2016). Introducción a los flujos de trabajo. Recuperado de <https://support.office.com/es-es/article/Introducci%C3%B3n-a-los-flujos-de-trabajo-15d0d459-cce8-4d8b-b34f-9ab6cda1075b#bm2>

OMG (2011). Business Process Modeling Notation Specification. Recuperado de <http://www.omg.org/spec/BPMN/2.0/PDF>

Plesums C. (2002). An Introduction to Workflow. Recuperado de [http://www.wfmc.org/information/introduction\\_to\\_workflow02.pdf](http://www.wfmc.org/information/introduction_to_workflow02.pdf)

Pressman, R. (2014). Software Engineering: A Practitioner's Approach. McGraw Hill, 8a. edición, EUA.

Prior, C. (2003). Workflow and Process Management. Maestro BPE Pty Limited, Australia. Recuperado de [http://www.wfmc.org/information/Workflow\\_and\\_Process\\_Management.pdf](http://www.wfmc.org/information/Workflow_and_Process_Management.pdf)

Sommerville I. (2015). Software Engineering. Pearson, 10a Edición, EUA.

## Notas biográficas:



**Víctor Hugo Menéndez Domínguez** Víctor Hugo Menéndez Domínguez es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha, España. Es Profesor Titular en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con repositorios de documentos digitales, la representación del conocimiento y la gestión de Objetos de Aprendizaje.



**María Enriqueta Castellanos Bolaños** María Enriqueta Castellanos Bolaños es Maestra en Gestión de Tecnología de Información por la Universidad Anáhuac Mayab, México. Es Profesora de tiempo completo en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con la gestión del conocimiento y la Ingeniería Web.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

# **Evidencia Empírica de la Minería de Procesos en la Implantación de CMMI-DEV**

## **Empiric Evidence of Process Mining in CMMI-DEV Implementation**

Paola E. Velazquez-Solis<sup>1</sup>  
paola.velazquez@uabc.edu.mx

Brenda L. Flores-Rios<sup>1</sup>  
brenda.flores@uabc.edu.mx

María Angélica Astorga-Vargas<sup>1</sup>  
angelicaastorga@uabc.edu.mx

Jorge E. Ibarra Esquer<sup>1</sup>  
jorge.ibarra@uabc.edu.mx

Félix Fernando González Navarro<sup>1</sup>  
feranando.gonzalez@uabc.edu.mx

Francisco J. Pino<sup>2</sup>  
fjpino@unicauca.edu.co

<sup>1</sup> Universidad Autónoma de Baja California Campus Mexicali, Mexicali, B.C., México

<sup>2</sup>Universidad del Cauca, Colombia

**Resumen:** La minería de procesos tiene como objetivo descubrir, monitorear y mejorar procesos a través del análisis de los diversos registros de eventos generados por los procesos de la organización. El objetivo de este trabajo es presentar la evidencia empírica de la inclusión estratégica de la disciplina de minería de procesos en proyectos de mejora de procesos de software implementados con CMMI. En el proceso de mapeo sistemático de la revisión de la literatura, se establecieron cuatro categorías para clasificar los hallazgos encontrados (Fundamentos teóricos, propuestas, herramientas y sistemas de información y algoritmos) para presentar los estudios que cumplen con el objetivo. Se concluye que la interdisciplinariedad de la minería de procesos con un modelo de referencia de procesos como CMMI-DEV apoya la implementación y evaluación de las áreas de procesos, al aplicar técnicas y algoritmos de minería de procesos que faciliten la exploración y explotación de los registros de eventos relacionados a la ejecución de las actividades almacenados en un repositorio.

**Palabras clave:** Minería de Procesos, Mejora de Procesos de Software, Registro de Eventos.

**Abstract:** The discipline of process mining aims to discover, monitor and improve processes through the analysis of the various event logs generated by the organization's processes. In such a context, the goal of this paper is to present empirical evidence of the strategic inclusion this discipline into software process improvement projects based on CMMI. In the process of mapping the results of a systematic literature review, four categories were established to classify the findings (Theoretical Foundations, proposals, tools and information systems and algorithms) and to present studies that achieve the objective established. We conclude that the interdisciplinary collaboration of process mining with a reference process model such as CMMI-DEV supports the implementation and evaluation of process areas, applying techniques and algorithms of process

mining that facilitate the exploration and exploitation of event logs related to the implementation of activities stored in a repository.

**Keywords:** Process Mining, Software Process Improvement, Event Log.

# 1. Introducción

En un proyecto de mejora de procesos de software (SPI por sus siglas de Software Process Improvement), en la medida que se genera la evidencia de la ejecución de las actividades o prácticas, requeridas por el modelo de referencia de procesos o ciclo de mejora, se incrementa la cantidad de datos e información por cada proceso, provocando algunos problemas en la escalabilidad o control de la información almacenada en el repositorio de evidencias de la organización. Por lo general, se utilizan como repositorios las bases de datos, bases de conocimiento, intranet o sistemas de información donde el tamaño se basa en criterios de capacidad y eficiencia de almacenamiento y no en su uso posterior, mucho menos en la exploración, explotación y análisis de los registros. Es así, como el limitado acceso a dicha información impide el enriquecimiento de criterios para la toma de decisiones necesarias en el descubrimiento de nuevas reglas de negocio, mejora continua, monitoreo o un control de los procesos ya establecidos (Orellana García et al., 2015).

Al revisar los diferentes estándares y modelos de procesos de software, como CMMI, ISO/IEC 15504, ISO/IEC 12207:2004, NMX-I-059-NYCE-2011 y COMPETISOFT (Pino et al., 2006), se detectó que no plantean técnicas, herramientas o algoritmos para extraer, de manera adecuada, la información o el conocimiento existente en sus registros (registro de eventos). Específicamente para dar evidencia de la implantación del modelo de referencia CMMI, se utiliza la revisión de la ejecución de las prácticas y artefactos almacenados en los repositorios, además de entrevistas dirigidas a los roles involucrados. De esta forma, los registros están compuestos por instancias del área de procesos y cada

instancia contiene la información relacionada con la ejecución de la práctica como nombre de la actividad, rol responsable y tiempo entre otros elementos. Siendo los registros de eventos una fuente de datos para el análisis de la madurez organizacional. En la representación escalonada se establecen cinco niveles de madurez (del 1 al 5) en función de un grupo de áreas de procesos (Chrissis, 2011). Desde el nivel 2 (Gestionado) se garantiza que las áreas de procesos se planifican y ejecutan de acuerdo con las políticas; los proyectos de software se conforman con personal calificado que dispone de recursos adecuados para producir resultados controlados (Chrissis, 2011). Al revisar el Maturity profile del 2015 de CMMI v1.3, el mayor número de evaluaciones corresponde a organizaciones con nivel 3 de madurez (Definido) (CMMI Institute, 2015), considerándolo el nivel que requiere más esfuerzo y conocimiento técnico, en función al mayor número de áreas de procesos asociados a las categorías que se evalúan dentro de éste.

Bajo este contexto, en este documento se expone la importancia de implementar técnicas o algoritmos que permitan ser adoptados o aplicados por las organizaciones de desarrollo de software para conocer el estado actual de sus procesos y/o diagnosticar el nivel de capacidad de los mismos en un tiempo determinado. Una estrategia es apoyarse de la disciplina de minería de procesos, la cual tiene como objetivo descubrir, monitorear y mejorar procesos a través de la extracción y exploración del conocimiento contenidos en los registros de eventos de los sistemas de información (van der Aalst, 2011; Orellana García et al., 2015). Los registros de eventos corresponden al histórico de ejecución de los procesos, donde se encuentran las instancias o casos del proceso (Orellana García et al., 2016). Cuando una organización cuente con un proyecto SPI y genere un volumen de información, la minería de procesos brindaría las técnicas, algoritmos o herramientas con el objetivo de llevar a cabo la exploración de los registros de eventos (Yzquierdo et al., 2013) con una mayor cobertura y confiabilidad, mientras se mantienen menores costos y esfuerzos (Valle et al., 2014). De esta forma, los procesos implementados, la información y el conocimiento explorado y descubierto, estarían perfectamente alineados a los

requerimientos relacionados con el cumplimiento de normas, eficiencia y servicios del cliente (van der Aalst et al., 2012).

Este documento proporciona un mapeo sistemático como una evidencia empírica dirigida a la comunidad de Ingeniería de software e investigadores de Inteligencia Artificial presentando la interrelación de la minería de procesos en proyectos SPI basados en CMMI. La técnica de mapeo sistemático define un proceso y un informe que permite categorizar los resultados que han sido publicados hasta el momento en un área determinada (Petersen et al., 2008). Los resultados son analizados según su aportación metodológica, casos de estudio y herramientas empleadas para la exploración de los registros de eventos. Así mismo, el documento se estructura de la siguiente manera: en la sección 2 se aborda el marco conceptual con información base sobre la minería de datos, minería de procesos y mejora de procesos de software. La sección 3 describe la metodología de utilizada en el mapeo sistemático de los trabajos de minería de procesos en proyectos SPI basados en CMMI; la sección 4 presenta los resultados, ventajas y desventajas identificados en cada uno de ellos. Por último, se presentan las conclusiones y trabajo futuro.

## 2. Marco Conceptual

La minería de procesos es una disciplina relativamente joven que las organizaciones están incorporando a sus proyectos y aplicaciones con el objetivo de mejorar sus procesos de negocio (Orellana García et al., 2015), por lo que es importante distinguir la diferencia entre la minería de datos y la minería de procesos. Mientras la primera es utilizada para encontrar patrones en grandes volúmenes de datos, en una relación de patrones y datos; la minería de procesos encuentra las relaciones de los procesos de la organización dentro de los datos, explorando la información de los registros; es decir, se sitúa entre la minería de datos y el análisis de procesos, implementando Business Intelligence (van der Aalst, 2011) y Business Process Management, que tiene como objetivo el

descubrimiento de modelos basados en el registro de eventos disponibles (Gupta, 2014).

En la siguiente sección, se presenta el estado del arte de registros de eventos, minería de procesos y mejora de procesos de software basados en CMMI-DEV.

## 2.1 Registro de Eventos

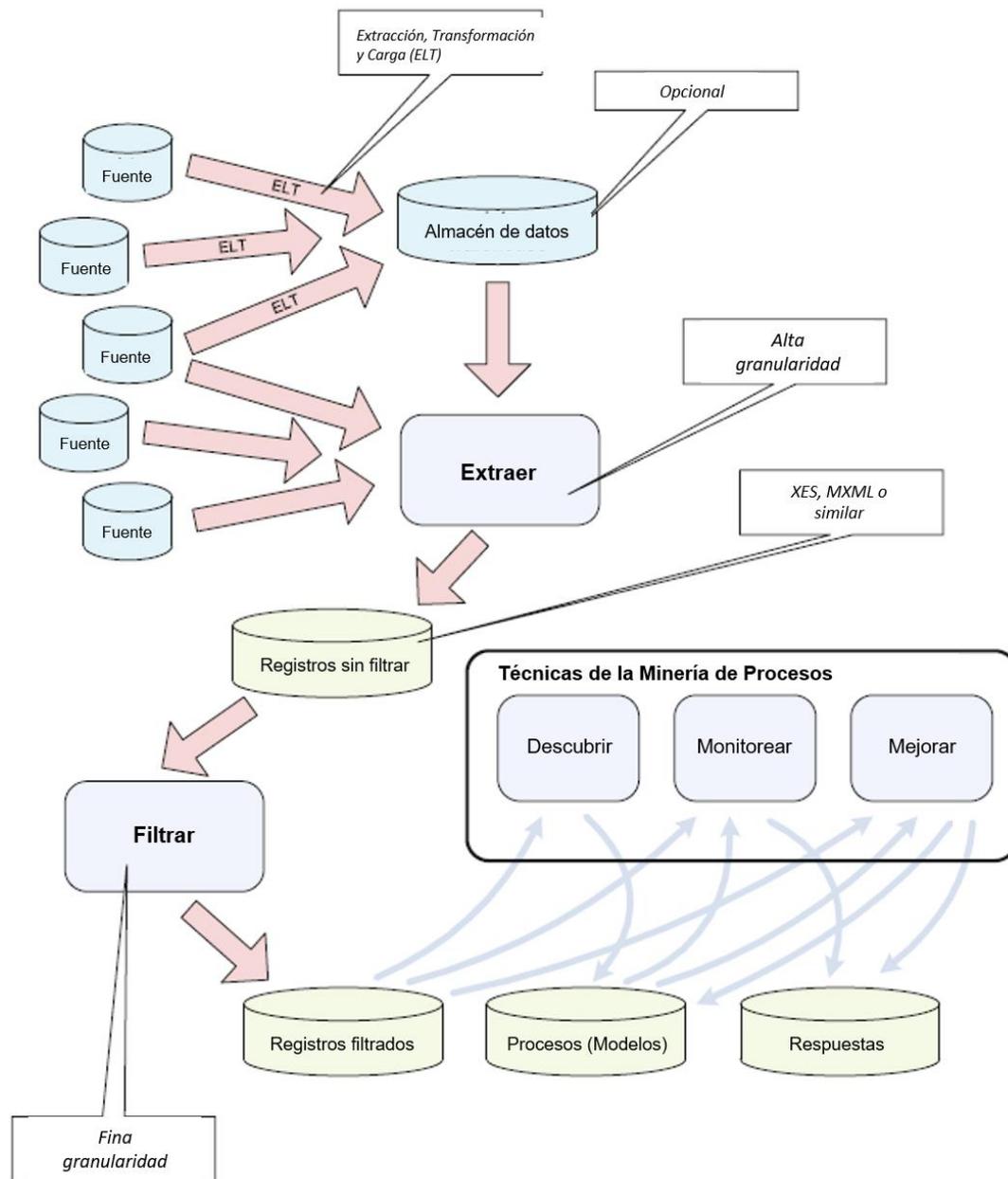
Los registros de eventos tienen como objetivo extraer información a partir de los datos almacenados en los registros de trazas de los sistemas de información, en busca de errores inconsistencias, vulnerabilidades y variabilidad en los procesos que se ejecutan (Orellana García et al., 2016). El propósito es analizar los datos de eventos desde la perspectiva orientada a procesos.

La Figura 1 muestra el flujo de trabajo de los registros de eventos, donde el punto de partida son los datos en bruto ocultos en todo tipo de fuentes de datos. Una fuente de datos puede ser un archivo plano simple, una hoja de cálculo de Excel, un registro de transacciones, o una tabla de base de datos (van der Aalst, 2011). Los datos de los registros de eventos, con mayor frecuencia, se dispersan típicamente a través de diferentes fuentes, requiriendo grandes esfuerzos para recopilar los datos pertinentes. Los eventos también pueden capturarse buscando en los intercambios de mensajes, el registro de lectura y escritura y también las fuentes de datos que son estructurados por metadatos. Desafortunadamente, en muchas situaciones, los datos son datos no estructurados (Verdonk et al., 2010).

En la misma figura, se observan flujos de salida desde las fuentes de datos que indican la leyenda ETL refiriéndose a la extracción, transformación y carga (ETL). ETL se utiliza para describir el proceso que consiste en: a) extracción de datos de fuentes externas, b) transformación de las necesidades operacionales, y c) la carga en el sistema de destino en un almacén de datos o base de datos relacional. Un almacén de datos es un repositorio único lógico de los datos transaccionales y operativos de una organización. El almacén de datos no

produce datos. El objetivo es unificar la información de tal manera que se puede utilizar para la presentación de informes, análisis, pronóstico, etc. Dependiendo de las preguntas y puntos de vista elegidos, diferentes registros de eventos pueden ser extraídos del mismo conjunto de datos, basándose en el registro filtrado, dependiendo de los diferentes tipos de minería de procesos que se deseen aplicar: descubrimiento, de conformidad y mejora (van der Aalst, 2011). Estas tres técnicas se explican más adelante.

Es importante señalar que aunque la Figura 1 no refleja la naturaleza iterativa de todo el proceso, es probable que los resultados de la minería de procesos desencadenen nuevas preguntas que a su vez, conduzcan a la exploración de nuevas fuentes de datos y extracciones de datos más detallados. Por lo general, se necesitan varias iteraciones de la extracción, filtrado y fases de la producción. Algunos sistemas de información PAIS (por las siglas de Processs Aware Information Systems) del tipo workflow, BPMS, ERP, CRM, entre otros (Pérez Alfonso et al., 2012) permiten generar automáticamente los registros de eventos, descubrir modelos de procesos considerando distintos objetivos o perspectivas como el control de flujo, organizacional, o temporal por medio de diversas representaciones (Pizarro Álvarez, 2013). De esta forma, los registros de eventos almacenan la información de los eventos asociados a las actividades de los procesos (Pérez Alfonso et al., 2012).



**Figura 1:** Diagrama de extracción de datos para un registro de eventos.  
Traducida de (Van der Aalst, 2011).

## 2.2 Minería de Procesos

El concepto de minería de procesos ha evolucionado a través de los años. En un inicio se definió como una técnica que utiliza los registros de flujo de trabajo

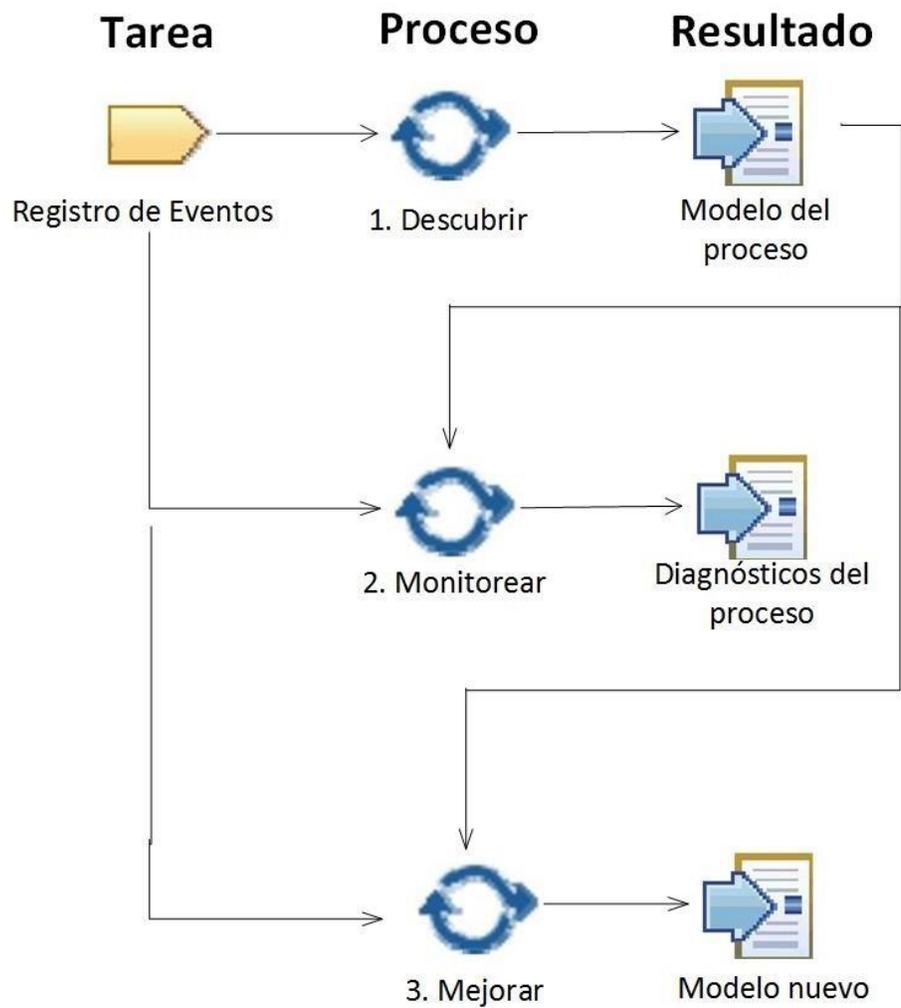
para descubrir el proceso que en realidad está ejecutándose (Weijters & van der Aalst, 2001). Posteriormente, con la creación de nuevos algoritmos para el análisis de los registros de eventos, la minería de procesos se definió como un conjunto de técnicas y procedimientos que trataban de reconstruir de forma automática el modelo observado en los registros de eventos (van der Aalst et al., 2004). Antes de convertirse formalmente como una disciplina ante la IEEE en el 2011, la minería de procesos se refería a un grupo de técnicas con el objetivo de descubrir, mejorar y monitorear los procesos reales a través de la extracción del conocimiento de un conjunto de ejecuciones reales de procesos (Nakatumba, 2009). Hoy en día, los registros de eventos ya están automatizados en algunos sistemas de información (van der Aalst, 2011; Orellana García et al., 2015).

El objetivo de la minería de procesos es responder a preguntas de la perspectiva operacional de los procesos. Por ejemplo: ¿Qué sucedió realmente en el pasado?, ¿Por qué sucedió?, ¿Qué es probable que suceda en el futuro?, ¿Cuándo y por qué las organizaciones y personas se desvían?, ¿Cómo controlar mejor un proceso?, ¿Cómo rediseñar un proceso para mejorar su rendimiento? Los autores (van der Aalst et al., 2011), especifican que la minería de procesos brinda las técnicas que ayudan a las organizaciones a:

1. Descubrir el modelo de ejecución real del proceso. Los algoritmos de minería de procesos cuando son aplicados al análisis de los registros de eventos, permiten generar el modelo real de un proceso, el cual es expresado en una red de Petri o en la notación BPMN (por sus siglas de Business Process Modeling Notation). Dicho modelo representa la situación real y no se basa en la documentación de cómo se ejecuta el proceso o en la percepción de los involucrados en el mismo.
2. Determinar si el proceso cumple con la reglamentación y los procedimientos documentados. Comparar el modelo obtenido con los procesos documentados para determinar el cumplimiento de estándares, protocolos, reglamentación y políticas de ejecución de un proceso.
3. Analizar la interacción del personal que ejecuta el proceso. Brinda la posibilidad de construir redes sociales del proceso para analizar la interacción y descubrir posibles bucles que demoran el proceso.
4. Descubrir cuellos de botella. Especificar actividades tipo cuello de botella.

- Determinar la relación entre las variables de un caso, por medio de la aplicación de técnicas de clasificación.

En la Figura 1 se indicó que las técnicas de minería de procesos asumen que es posible registrar eventos de manera secuencial, de tal manera que cada uno de los eventos se refiere a una actividad; es decir, un paso bien definido en algún proceso y a su vez éste se relacione a un caso particular, como una instancia del proceso. En la Figura 2 se presenta un diagrama para estructurar el flujo de actividades para la implementación de las técnicas de: Descubrir, Monitorear y Mejorar.



**Figura 2:** *Técnicas de Minería de Procesos. Traducida y adaptada de (van der Aalst, 2011).*

En la Figura 2 se muestra cómo la primera técnica de minería de procesos está basada en el descubrimiento. Esta es una de las técnicas más destacadas (van der Aalst, 2011), la cual consiste en tomar como entrada un registro de eventos y generar un modelo del proceso sin usar ninguna información a-priori. La segunda técnica de minería de procesos se basa en el monitoreo y la conformidad. Es aquí donde se compara un modelo que ya existe contra un registro de eventos del mismo proceso (van der Aalst et al., 2012). La verificación de conformidad puede ser usada para asegurarse si la realidad del diagnóstico obtenido coincide con los registros de eventos y el modelo del proceso, y viceversa. Mientras que la segunda técnica busca medir dicha conformidad, el tercer tipo de técnica de minería de procesos busca mejorar o extender un modelo ya existente (Wang et al., 2013). El objetivo es extender o mejorar un modelo existente utilizando la información acerca de la ejecución de un proceso real almacenada en algún registro de eventos. La Figura 2 presenta una relación entre la información requerida para la implementación de las tres técnicas de minería de procesos, así como la salida de una puede alimentar la entrada de otra.

### 2.3 Mejora de Procesos de Software en CMMI

Un programa SPI permite orientar a las organizaciones en la implementación, evaluación y mejora de sus procesos (Niazi et al., 2005). El principal enfoque de SPI está basado en el modelo, el cual compara los procesos de la organización con un modelo de referencia de procesos y utiliza una evaluación SPI para identificar debilidades y posibles opciones de mejora. Un ejemplo del enfoque SPI basado en modelo es CMMI (Vasconcellos et al., 2017). SPI basado en CMMI significa una estrategia formal para las organizaciones de software en lograr una implementación exitosa del modelo (Coleman & O'Connor, 2008) y aportar importantes beneficios para ellas, como el incremento de la satisfacción

del cliente, la eficiencia del proceso de desarrollo y la definición y cumplimiento de los objetivos de calidad (Durón del Villar & Muñoz-Mata, 2013).

Para la implementación de un programa SPI las etapas a realizar son: 1) Diagnóstico de procesos actual de la organización en función de los requisitos normativos a través de la evaluación y la correspondencia de los procesos de la organización con el modelo de referencia; 2) Establecimiento de la estrategia de implantación con base en las necesidades de mejora identificadas en la evaluación diagnóstico y la definición del número de ciclos de mejora para cumplir con el nivel de madurez de la organización definido en su alcance; y 3) Evaluación de los resultados obtenidos y mejora de los procesos implantados (Flores Rios et al., 2014). En este caso, el método de evaluación de CMMI es conocido como SCAMPI (por las siglas de Standard CMMI Appraisal Method for Process Improvement).

Existen organizaciones que aún enfrentan desafíos en la implementación de SPI basados en CMMI (Niazi, 2015), entre ellos la necesidad de contar con herramientas de software como un soporte valioso para el equipo de implementación, evaluación o equipo de mejora permitiéndoles recopilar, registrar clasificar y evaluar la evidencia sobre todo cuando el volumen y complejidad de la información son significativos (NYCE, 2010; Astorga et al., 2014). Otro desafío para las organizaciones es el mantener actualizados sus procesos de acuerdo a los flujos de trabajo reales con los que operan (van der Aalst, 2011).

Desde la perspectiva de la minería de procesos, los flujos de trabajo están incluidos implícitamente en cada registro de evento (actividad) asociado a cada uno de los procesos de la organización, permitiendo de esta manera realizar el descubrimiento, la mejora y el monitoreo de los procesos a través de la aplicación de herramientas de software para el análisis de datos dependiendo del ciclo de mejora y/o el nivel de madurez al que desee llegar la organización de software.

## 3. Metodología

Con el objetivo de proporcionar evidencia empírica que muestre los trabajos publicados hasta el momento sobre la incidencia de la disciplina de la minería de procesos en proyectos SPI basados en CMMI, en esta sección se describen las fases del mapeo sistemático como etapa previa a la revisión sistemática de la Literatura propuesta por Kitchenham (2007). La técnica de mapeo sistemático permite categorizar los resultados que se publican en una determinada área de interés (Martínez et al., 2011). El modelo de referencia de procesos CMMI se ha sostenido por más de dos décadas en la industria de software, tomándolo como referencia para la generación de otros estándares o modelos como NMX-059-NYCE-2011 y COMPETISOF (Oktaba, 2008). Así mismo, se ha probado que CMMI puede ser implementado no sólo en grandes organizaciones sino también por Pequeñas y Medianas Empresas (PyMEs) (Oktaba, 2008; Lester et al., 2010) otorgándoles una ventaja competitiva en el mercado internacional. México, desde el 2002 ha venido impulsando la estrategia de implantación de CMMI en PyMES, contando actualmente con 180 centros de desarrollo certificados en CMMI-DEV (Secretaría de Economía, 2016).

### 3.1 Preguntas de investigación

Esta fase se centra en la identificación de los hallazgos en la adopción de minería de procesos como soporte a proyectos SPI. Se han definido las siguientes preguntas de investigación:

- a) ¿Existen publicaciones sobre la implantación de proyectos SPI usando la minería de procesos? y
- b) ¿Cuáles son los principales hallazgos de la adopción de minería de procesos en proyectos SPI basados en CMMI?
- c) ¿Qué tipo de calidad de estudios se han publicado para reportar los resultados de la adopción de la minería de procesos en proyectos SPI basados en CMMI?

La primera pregunta se refiere a la identificación de evidencia sobre la minería de procesos en la implantación de un proyecto SPI independientemente del modelo para llevar a cabo la mejora de procesos o un modelo de referencia de procesos o los métodos de evaluación de procesos a seguir. Las siguientes preguntas hacen hincapié en la identificación de hallazgos, las ventajas y desventajas y tipos de calidad de estudios en la adopción de la minería de procesos específicamente con el modelo CMMI.

El objetivo es buscar referencias y publicaciones sobre la minería de procesos, sus técnicas y algoritmos utilizados según el tipo de estudio o clasificación (Fundamentos teóricos, Propuestas, Algoritmos, Sistemas de información) que han apoyado a un proyecto SPI basado en CMMI. Esta búsqueda se realiza con el apoyo de principales fuentes de información con enfoque técnico-científico, bajo los términos o cadenas de búsqueda claves de minería de procesos y procesos de software, durante el periodo 2005-2015.

### 3.2 Ejecución de la búsqueda

La búsqueda de literatura se realizó en cuatro fuentes electrónicas: Google Scholar, Researchgate, Scopus y Springer. El mapeo sistemático permite identificar, evaluar, interpretar y sintetizar todas las investigaciones existentes y relevantes en un tema de interés particular (Petersen et al., 2008). Las publicaciones más recientes dentro del período de búsqueda seleccionado en revistas especializadas, actas de conferencias e Internet (Kitchenham, 2007). Se utilizó una estructura de búsqueda, basado en conectores de palabras clave y lógica, obteniendo la siguiente cadena de búsqueda: "Process Mining" AND "Software Process Improvement".

En la primera búsqueda, se obtuvieron 18,641 documentos los cuales especifican la definición, los objetivos y propósitos de la minería de procesos. Durante 2001 a 2005, los autores (Weijters & van der Aalst, 2001; van der Aalst et al., 2007; Weijters, 2003), dieron las primeras propuestas sobre la minería de

procesos como una herramienta y/o técnica para analizar cualquier proceso en general. Se encontraron casos de estudio sobre la implementación de la minería de procesos en procesos hospitalarios, sistemas industriales y dependencias gubernamentales, reportando que: a) en los sistemas hospitalarios se permitió monitorear la utilización de los recursos, como los implementos quirúrgicos, detectar funcionamientos anómalos, como la desviación, el exceso en el tiempo de estancia de un producto o insumo en el almacén o su ausencia, las variaciones entre la realidad y lo prescrito (Orellana García et al., 2015); b) en los sistemas industriales ha permitido analizar la tramitación de las facturas enviadas por los distintos subcontratistas y proveedores (Verdonk et al., 2010); c) en auditorías en instituciones gubernamentales se logró encontrar violaciones de las normas específicas impuestas por la ley, indicar posibles fraudes, negligencias, riesgos, o ineficiencias (van der Aalst, 2011).

Para segmentar los documentos que asociaran a la minería de procesos con SPI se refinó la cadena de búsqueda “process mining” AND “software process improvement”, obteniendo como resultado 897 documentos. La cadena de búsqueda se adaptó a cada biblioteca electrónica y se restringieron las coincidencias en título, resumen y palabras clave. En el transcurso del 2015, el mayor número de evaluaciones de CMMI-DEV v1.3 corresponden con el nivel de madurez 3 (Definido), lo que indicaría ser el nivel que requiere más esfuerzo, tomando en cuenta el mayor número de áreas de procesos asociadas que se evalúan, sin considerar los niveles inferiores (CMMI Institute, 2015). Se consideraron las publicaciones que incluían las palabras clave: "software process improvement" OR "CMMI-DEV" OR "SCAMPI".

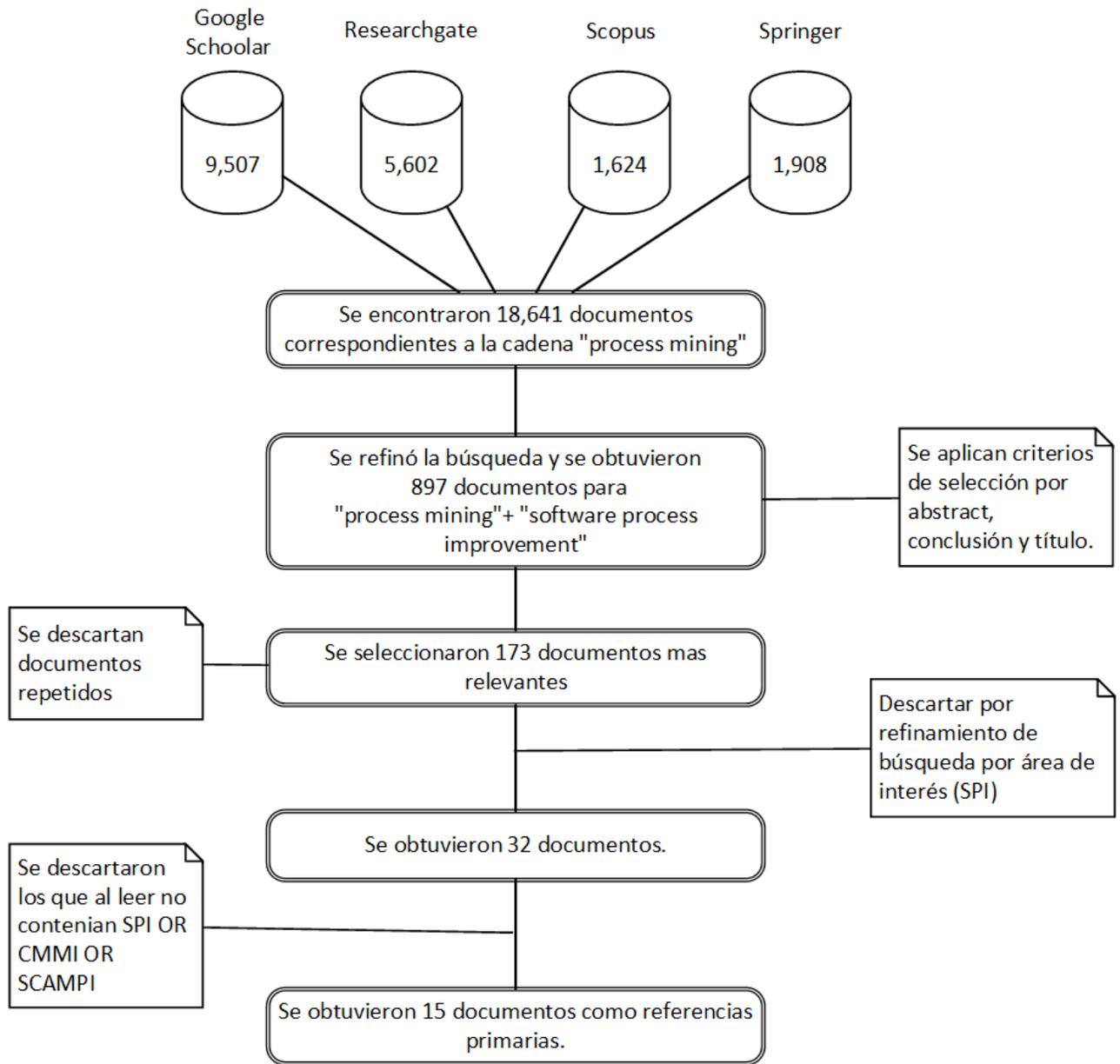
### 3.3 Filtrado de estudios

Con el fin de identificar los estudios más relevantes, se utilizó el siguiente procedimiento:

1. Selección de las palabras clave de la investigación;

2. Observación en cada biblioteca digital, tratando con las palabras clave de la investigación basadas en los criterios de inclusión y exclusión. Para la selección de los estudios primarios se definieron los siguientes criterios de inclusión: documentos, informes técnicos, artículos técnicos-científicos que presenten la inserción de la minería de procesos en SPI. Los trabajos seleccionados tuvieron que ser publicados entre 2005 y 2015 en revistas y conferencias, escrito en idioma Inglés. Se excluyeron aquellos resultados donde sólo se tenía acceso al resumen.
3. Análisis de cada documento a través del título y el resumen;
4. Descarga de documentos que cubren los criterios de búsqueda;
5. Análisis de la introducción y las conclusiones a conciencia, y teniendo una visión rápida del resto de las secciones en un documento. Se aplicó este proceso en cada biblioteca electrónica.

Con base en criterios de resumen y título, se aplicó un nuevo filtrado de los estudios reduciendo el número a 173 documentos. Se descartaron los que no consideraban los términos de SPI, CMMI y SCAMPI; así como los documentos que contenían información repetida. Finalmente, se obtuvo un total de 15 documentos (Figura 3). En 11 documentos de ellos, se plantea la interdisciplinariedad de la minería de procesos aplicada a proyectos SPI. En los 4 restantes, se enfocan específicamente al modelo de referencia de procesos CMMI utilizando la minería de procesos como una herramienta para la exploración de la información. Este mapeo sistemático permitió observar que del conjunto seleccionado de 15 documentos finales el 26.66% está relacionado con el objetivo de esta investigación.



**Figura 3:** Proceso de ejecución de la búsqueda y filtrado de resultados.  
Elaboración propia.

### 3.4. Esquema de clasificación

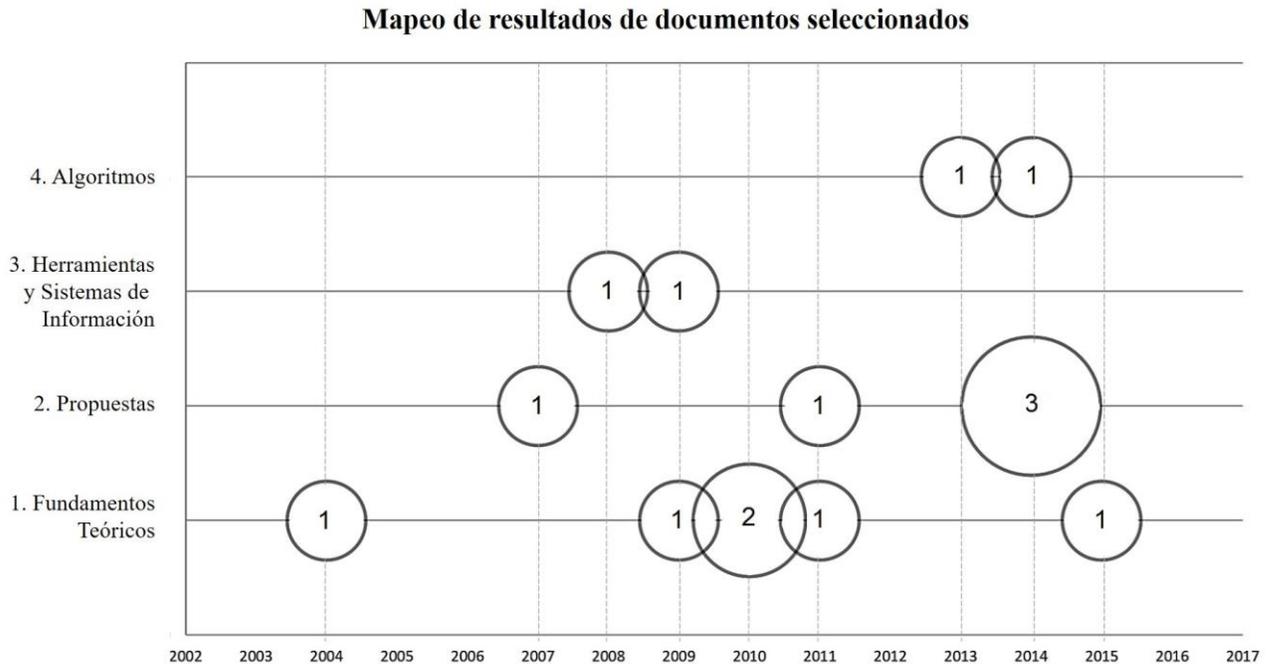
La Figura 3 muestra los 15 documentos relevantes seleccionados los cuales fueron clasificados en: 1) Fundamentos teóricos, 2) Propuestas, 3) Herramientas y Sistemas de Información y 4) Algoritmos. Partiendo de que desde hace 4 años

la minería de procesos es una disciplina formal, la categoría 1) Fundamentos teóricos se relaciona con el marco conceptual base para los investigadores y profesionales. La categoría 2) Propuestas contribuye a responder la pregunta de investigación 2 presentada en la sección 3.1. Esta categoría muestra los principales hallazgos de la adopción de la minería de procesos en proyectos SPI, específicamente basados en CMMI. Los registros de eventos se han explotado con ayuda de herramientas disponibles y se han desarrollado varios sistemas académicos y comerciales (van der Aalst & Van Dongen, 2013) por tal motivo, la categoría 3 se relaciona a la identificación y caracterización de herramientas y sistemas de información. Por último, la categoría 4 se refiere a los algoritmos o tipo de técnicas de minería de procesos que apoyan en gran medida a la resolución de la problemática en una organización de desarrollo de software en la exploración y explotación del conocimiento y registro de eventos brindando así la evidencia de la ejecución de actividades en un proyecto SPI implementando CMMI-DEV.

### 3.5 Mapeo

El último paso del mapeo sistemático consiste en generar un modelo (mapa visual) de las dimensiones. La Figura 4 muestra el diagrama de burbuja que visualiza la frecuencia de documentos por años de publicación y la intersección de cada uno de los documentos con su clasificación asignada. Esto significa que el tamaño de una burbuja es proporcional al número de documentos reportados en el año y en relación a una única categoría. Por ejemplo, la categoría 2) Propuestas visualiza la burbuja con mayor tamaño con un total de 3 documentos en el 2014. Otra vista es la frecuencia de documentos publicados por años, donde la categoría 1) Fundamentos teóricos cuenta con 6 documentos en un periodo del 2004 al 2015, concentrándose más en el periodo del 2009 al 2011. De este modo, las dos primeras categorías reflejan, de manera visual, un mayor esfuerzo en establecer los fundamentos teóricos y las propuestas sobre la minería de procesos en proyectos SPI implementando el modelo de referencia CMMI-DEV; a diferencia de las categorías 3 y 4 con las menores frecuencias.

De igual forma, en la Figura 4 se observa que a partir del 2007 la frecuencia de los documentos va en aumento por año de publicación, siendo el año de 2014 el de mayor número con 4 documentos relacionados a las categorías de 2) Propuestas y 4) Algoritmos.



**Figura 4:** Visualización del mapeo sistemático en diagrama de burbuja.  
Elaboración propia.

## 4. Análisis y Discusión

Durante 15 años, los aportes más completos para los principios de la minería de procesos fueron hechos por Wil M. P. van der Aalst (2001, 2003, 2004, 2007, 2011, 2012, 2013). En la Tabla I se presenta una relación de definiciones del término de minería de procesos, sugeridas por diversos autores, las cuales han ido en evolución al incluir los análisis de los registros de eventos y las técnicas, algoritmos y herramientas basadas en principios de minería de datos, Business Intelligence y Business Process Management. Actualmente, la minería de procesos es reconocida como una disciplina ante la IEEE (van der Aalst et al., 2012).

**Tabla 1.**Relación de propuestas para la definición de minería de procesos

Año	Autor	Definición
2001	van der Aalst	Técnica que utiliza los registros de flujo de trabajo para descubrir el proceso que en realidad está ejecutándose (van der Aalst, 2001).
2003	Weijters	Conjunto de técnicas y procedimientos que tratan de reconstruir de forma automática el modelo observado en los ficheros de registros (Weijters, 2003).
2009	Nakatumba & van der Aalst	Conjunto de técnicas que tienen como principal objetivo, descubrir, mejorar y monitorear los procesos reales a través de la extracción de conocimientos de un conjunto de ejecuciones reales de procesos (registro de eventos) (Nakatumba & van der Aalst, 2009).
2012	van der Aalst et al., 2012	Disciplina permite descubrir, monitorear y mejorar los procesos identificados en los sistemas de información que poseen las organizaciones a partir de sus registros de eventos (van der Aalst et al., 2012).
2014	Gupta	Disciplina que construye el puente entre la minería de datos como un enfoque de Business Intelligence y Business Process Management y tiene como objetivo el descubrimiento de modelos basados en el registro de eventos disponibles (Gupta, 2014).
2015	Aguirre	Disciplina que está compuesta por una serie de herramientas y técnicas basadas en la minería de datos para analizar procesos de negocio cuyo registro de eventos de ejecución real se encuentran disponibles en sistemas de información (Aguirre, 2015).

El acervo referencial obtenido por el mapeo sistemático permitió identificar que la categoría 1) Fundamentos teóricos tuviera la mayor concentración de aportaciones conceptuales de manera general (Figura 3), y específicas tal como se visualiza en el eje y de la Figura 4. Por otro lado, en la categoría 2)

Propuestas, se relaciona la integración de la minería de procesos con sistemas de información, modelos de referencia de procesos (Rubin, 2007; Samalikova, 2014; Valle et al., 2014b), modelos de evaluación de procesos (Gruañas, 2011; Valle et al., 2014), monitoreo y control de procesos.

La categoría 3) Sistemas de Información y herramientas agrupa las diversas técnicas y algoritmos de minería de procesos en herramientas de software comerciales: ARIS Process Performance Manager por Software AG, Disco por Fyxicon (Papadimitriou & Sun, 2008), Enterprise Visualization Suite por Businesscape, Interstage BPME por Fujitsu, Process Discovery Focus por Iontas, Reflect one por Pallas Athena y Reflect por Futura Process Intelligence. Por el contrario, la herramienta de minería de procesos de código abierto ProM es ampliamente utilizada en todo el mundo y proporciona un punto de partida sencillo para los profesionales, estudiantes y académicos (Van Der Aalst & Van Dongen, 2013).

En dichos sistemas de información se encontró que de acuerdo a la categoría 4) Algoritmos, se implementan principalmente tres algoritmos básicos: heurístico, difuso y genético (Gupta, 2014). La minería heurística proporciona una vista de flujos de trabajo científico, considerando la dependencia de larga distancia. Este algoritmo obtiene el modelo del proceso expresado en una C-net, es robusto ante el ruido y las ejecuciones incompletas de los procesos (Puldon & Llanes, 2005). El algoritmo difuso brinda una vista amplia del flujo de trabajo de forma científica mediante el control para mostrar una tarea diferente según el nivel de importancia. El algoritmo difuso devuelve el modelo de proceso en una red de Petri, con la limitante de no ser robusto ante el ruido, procesos poco estructurados o ejecuciones incompletas (Gupta, 2014). El tercer algoritmo trabaja con minería genética siendo capaz de obtener un modelo del proceso que refleja adecuadamente la realidad expresada en el registro de eventos pero tiene un alto costo computacional (Puldon & Llanes, 2005). El algoritmo genético proporciona una vista de la frecuencia en que se ejecutan las tareas y la sucesión entre las mismas. Así mismo, este algoritmo descubre todas las estructuras de

control de flujo comunes (Gupta, 2014). El modelo extraído mediante cualquiera de los tres algoritmos de minería de procesos puede brindar una vista del(los) proceso(s) en un determinado punto de la ejecución; esto conduce a buscar diversas formas de analizar el proceso para la manipulación de los cambios en él.

En relación a las implicaciones de la minería de procesos con respecto a CMMI, Valle et al. (2014, 2014b) desarrolló un marco (estructura y contenido) para aplicar técnicas de minería de procesos en las evaluaciones SCAMPI con el objetivo de llevarla a cabo con una mayor profundidad y cobertura, mientras se mantiene el mismo nivel de esfuerzo. Los autores indican que la minería de procesos facilitaría el proceso de evaluación tipo SCAMPI, brindando la información que se requiere sobre las áreas de procesos definidas para un modelo de referencia de procesos, tal como CMMI-DEV v1.3.

El autor Samalikova (2014), realizó un estudio para identificar los elementos que apoyan las evaluaciones de procesos de software con CMMI. Identificó los elementos clave dentro de los objetivos y prácticas genéricas, encontrando en la minería de procesos una forma de apoyar los enfoques actuales para la evaluación y mejora de procesos de software. De igual manera, Gruañas (2011) investigó el potencial de la minería de procesos para apoyar un proyecto SPI y evaluación de procesos de software basado en CMMI. Propuso una metodología concreta para encontrar las relaciones entre la minería de procesos y la evaluación de dos áreas de procesos de CMMI para un nivel 3 de madurez: Enfoque en Procesos de la Organización (OPF) y Análisis de Decisiones y Resolución (DAR). Concluyendo con que la minería de procesos es un camino viable para la recolección de datos en una evaluación y de esta forma reconstruir los procesos que se modelan con CMMI.

De esta forma, las aportaciones de (Samalikova, 2014; Valle et al., 2014, Valle et al., 2014b; Gruañas, 2011) recomiendan aprovechar, de forma significativa, los datos de eventos o registros de eventos en un proyecto SPI basado en CMMI-

DEV permitiendo que los procesos de monitoreo y control de la ejecución de la implantación de las áreas de procesos, la gestión de proyectos, gestión de procesos y el de evaluación SCAMPI, por mencionar algunos se apoyarían de un modelo para la exploración y explotación del conocimiento asociado al cumplimiento de las áreas de procesos establecidos para un nivel específico de madurez y/o mejorar el modelo del proceso.

## 5. Conclusiones

Los hallazgos encontrados en el mapeo sistemático demuestran una evolución de la minería de procesos durante el periodo seleccionado, reconociéndose inicialmente como una técnica hasta llegar a formalizarse como una disciplina, con la inclusión de técnicas, algoritmos y herramientas basadas en principios de minería de datos, que permiten la exploración y explotación de información a partir de los registros de evento. Así mismo, se encontró evidencia entre la Minería de Procesos en proyectos SPI que han tenido como propósito apoyar los enfoques actuales de implementación y evaluación de procesos de software definidos por el modelo de referencia de procesos CMMI-DEV y su método de evaluación SCAMPI.

La evidencia encontrada en las cuatro categorías (Fundamentos teóricos, propuestas, herramientas y sistemas de información y algoritmos) representa una base sobre la cual las organizaciones desarrolladoras de software encuentren una solución a los problemas que enfrentan asociados a la complejidad en el manejo de gran cantidad de información o a la falta de experiencia por parte del equipo de mejora en el uso de técnicas o algoritmos para ello. La implicación de la minería de procesos en este tipo de organizaciones es una alternativa para obtener una mayor visibilidad en la ejecución de las actividades requeridas por los modelos de procesos, a partir de la información almacenada en sus repositorios y concentrarse en aquellas que ayuden a satisfacer los objetivos del negocio permitiéndola ser más competitiva en la industria de software.

Actualmente, se está trabajando en un proyecto en el que se aplicarán los elementos conceptuales encontrados en esta revisión en una empresa mexicana certificada en CMMI-DEV con un el nivel 3 de madurez, para analizar los registros de eventos generados durante un año en las áreas de proceso de gestión de proyectos básicas, y de esta forma obtener un modelo del proceso que se compara con las evidencias requeridas en el proceso de evaluación SCAMPI.

## Agradecimientos

Este proyecto se encuentra registrado en la Coordinación de Posgrado e Investigación de la Universidad autónoma de Baja California por medio del programa 111/1298. Así mismo, se desea agradecer al CONACYT por el financiamiento otorgado al primer autor para realizar una estancia de investigación en la Universidad del Cauca (UNICAUCA)

## Referencias

Aguirre, S. (2015). Metodología para la aplicación de minería de procesos. Tesis Doctoral. Colombia.

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). CMMI for Development®: Guidelines for Process Integration and Product Improvement. 3rd Edition, Addison-Wesley Professional.

CMMI Institute. (2015). Maturity Profile June 30 2015. <http://cmmiinstitute.com/resources/process-maturity-profile-july-2015>, Octubre 2015.

Coleman G. & O'Connor R. (2008). Investigating software process in practice: a grounded theory perspective. *Journal of Systems and Software*, 81, pp. 772–784.

Durón del Villar, B. L. & Muñoz-Mata, M. A. (2013). Selección de estrategias para la implementación de mejoras de procesos de Software. *ReCIBE*, 2, pp. 15.

Cruañas, J. R. (2010). "Process Mining Opportunities for CMMI Assessments" (Doctoral dissertation, Universitat Politècnica de Catalunya. Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.

Flores Rios B.L., Astorga Vargas M.A, Rodríguez Elias O.M, Ibarra Esquer J.E. & Andrade M.C. (2014). Interpreting the mexican standards for software process implementation and capacity assessment under a knowledge management approach. *Revista de la Facultad de Ingeniería. Universidad de Antioquia*, 71,pp. 85-10.

Gupta, E. P. (2014). Process Mining a Comparative Study. *International Journal of Advanced Research in Computer and Communications Engineering*, 3, p. 5.

Kitchenham, B. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report, Software Engineering Group, Department of Computer Science, University of Durham.

Lester N. G., Wilkie F. G., McFall D., Ware M. P. (2010). Investigating the role of CMMI with expanding company size for small-to medium-sized enterprises. *Journal of Software Maintenance and Evolution: Research and Practice*, 22, pp. 17-31.

Martínez, Y., Castro, C. C., & Beigbeder, S. M. (2011). Evidencia empírica sobre mejoras en productividad y calidad en enfoques MDD: un mapeo sistemático. *REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software*, 7(2), 6-27.

Nakatumba, J., & van der Aalst, W. M. (2009). Analyzing resource behavior using process mining. In *International Conference on Business Process Management*. Springer Berlin Heidelberg, pp. 69-80.

Niazi M. (2015). A comparative study of software process improvement implementation success factors. *Journal of Software: Evolution and Process*, 27 (9),pp. 700–722.

Niazi, M., Wilson, D. & Zowghi, D. (2005). A maturity model for the implementation of software process improvement: An empirical study. *Journal of Systems and Software*, 74(2), pp.155-172.

NYCE (2010). NMX-I-15504/03-NYCE-2010.Tecnología de la Información – Evaluación de los Procesos –Parte 03, Guía para realizar una evaluación. México.

Oktaba H. & Piattini M. (2008). *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*. IGI Global research collection.

Orellana García, A., Larrea Armenteros, O. U., & Pérez Alfonso, D. (2015). *Generador de Registros de Eventos para el Análisis de procesos en el Sistema*

de Información Hospitalaria xavia HIS. Convención Salud. Disponible en <http://www.convencionsalud2015.sld.cu/index.php/convencionsalud/2015/paper/view/380/559>

Orellana García, A., Pérez Alfonso, D., & Estrada Sentí, V. (2016). Revisión de los principales modelos para aplicar técnicas de Minería de Procesos (Review of models for applying process mining techniques). *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología*, 4(1).

Papadimitriou, S., & Sun, J. (2008). Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining. In 2008 Eighth IEEE International Conference on Data Mining (pp. 512-521). IEEE.

Pérez Alfonso, D., Yzquierdo Herrera, R., Silverio Castro<sup>1</sup>, R., & Lazo Cortes, M. (2012). Utilización de minería de proceso en el entorno empresarial cubano. 12 Ed.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In 12th international conference on evaluation and assessment in software engineering, 17(1).

Pino, F., García, F., & Piattini, M. (2006). Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 2(1), pp. 6-23.

Pizarro Álvarez, G. E. (2013). Enfoque OLAP para el descubrimiento interactivo en minería de procesos. Tesis de Magíster en Ciencias de la Ingeniería. Pontificia Universidad Católica de Chile, 2013.

Puldon, J. J. & Llanes, R. S. (2005). Técnicas del algoritmo a y Minería Genética para la obtención de modelos de procesos a partir de trazas de ejecución de procesos. 7mo congreso de Iberoamericano de Ingeniería Mecánica.

Samalikova, J., Kusters, R. J., Trienekens, J. J. M., & Weijters, A. J. M. M. (2014). Process mining support for CMMI-based software process assessment, in principle and in practice. University of Technology Eindhoven, The Netherlands.

Secretaría de Economía (2016). Centros de Desarrollo Certificados/Verificados Vigentes en Modelos de Calidad. Disponible en [https://prosoft.economia.gob.mx/doc/PADRON\\_CENTRO%20DE%20DESARROLLO%20VIGENTE\\_2016\\_abr-18.pdf](https://prosoft.economia.gob.mx/doc/PADRON_CENTRO%20DE%20DESARROLLO%20VIGENTE_2016_abr-18.pdf)

Valle, A., Loures, E. R., & Portela, E. (2014). Process Mining Extension to SCAMPI. In SIMPDA (pp. 179-183).

Valle, A., Portela, E., Loures, E. R., & Deschamps, F. (2014b). Structure and Content of a Framework to Apply Process Mining Techniques in Scampi Assessments. In Proceedings of the International Annual Conference of the American Society for Engineering Management, p. 1.

Van der Aalst, W. M. P., (2011). Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg.

Van der Aalst, W. M. P., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, 16 (9), pp. 1128-1142.

Van der Aalst, W. M. P., Reijersa, A., Weijtersa, B., Van Dongena, A., Alves de Medeirosa M. S. & Verbeeka. (2007). Business process mining: An industrial application. Elsevier. 32(5), pp. 713–732.

Van der Aalst, W. M. P., Adriansyah, A., de Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., van den Brand, P., Brandtjen, R., Buijs, J., et al.(2012). Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011 Workshops, Part I. LNBP, Springer, Heidelberg. pp. 169–194.

Van Der Aalst, W. M., & Van Dongen, B. F. (2013). Discovering petri nets from event logs. In Transactions on Petri Nets and Other Models of Concurrency VII, Springer Berlin Heidelberg. pp. 372-422.

Vasconcellos, F., Landre G., Cunha J.A., Oliveira J.L., Ferreira R. A. & Vincenzi Auri M.R. (2017). Approaches to strategic alignment of software process improvement: A systematic literature review. The Journal of Systems and Software, 123, pp. 45–63.

Verdonk M., van Werf J., van der Aalst, W. M. P., van Hee K. M. (2010). Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. Computer, 43(3), pp. 90-93.

Wang J., Wong R. K., Ding J., Guo Q. & Wen L. (2013). Efficient Selection of Process Mining Algorithms. IEEE Transactions on Services Computing, 6(4), p.13.

Weijters, A. J. M. M., (2003). Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering, 10(2), pp. 151-162.

Weijters, A. J. M. M., & van der Aalst, W. M. P. (2001). Process mining: discovering workflow models from event-based data. In Belgium-Netherlands Conference on Artificial Intelligence.

Yzquierdo Herrera, R. (2013). Minería de proceso como herramienta para la auditoría. Ciencias de la Información. 44(2), pp. 25-32.

## Notas biográficas:



**Paola Elvira Velazquez Solis** es Ingeniero en Computación de la UABC y actualmente, estudiante de Maestría en Ciencias en el Instituto de Ingeniería de la UABC. Su área de investigación es la Minería de Procesos. Se ha desempeñado como analista de sistemas, analista de procesos y consultor en el sector privado de México.



**Brenda Leticia Flores Rios** es Doctora en Ciencias por la UABC. Responsable del área de Ingeniería del conocimiento del Instituto de Ingeniería, desarrollando proyectos de investigación relacionados a la Gestión del Conocimiento y Mejora de procesos de software. Imparte docencia en licenciatura y posgrado. Pertenece a la Red Temática Mexicana de Ingeniería de Software y a la Academia Mexicana de Computación. Ha participado como consultora en la implementación de la NMX-I-059-NYCE-2011 y como Miembro de Equipo Evaluador en SCAMPI A de CMMI-DEV nivel 3.



**María Angélica Astorga Vargas** es Maestra en Ciencias en el área de Computación. Profesora titular de tiempo completo en el Programa Educativo de Licenciado en Sistemas Computacionales de UABC. Su área de investigación es Modelos de Procesos de Software y Gestión del Conocimiento. Actualmente, se encuentra en formación de Doctorado. Pertenece a la Red Temática Mexicana de Ingeniería de Software. Ha participado como consultora en la implementación de la NMX-I-059-NYCE-

2011 y como Miembro de Equipo Evaluador en SCAMPI A de CMMI-DEV nivel 3.



**Francisco J. Pino** es Doctor en Ingeniería Informática por la Universidad de Castilla-La Mancha (España). Ingeniero en Electrónica y Telecomunicaciones, y Especialista en Redes y Servicios Telemáticos de la Universidad del Cauca (Colombia).

Profesor titular adscrito a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Miembro del Grupo IDIS (Investigación y desarrollo en ingeniería de software) de la Universidad del Cauca. Investigador Senior reconocido por Colciencias. Consultor en calidad y mejora de procesos, productos y servicios de software. Auditor Jefe por AENOR de ISO 15504-SPICE. Sus intereses de investigación y profesionales se enfocan en a área de calidad y mejora de procesos de desarrollo de software en pequeñas empresas y en entornos multimodelos. Sobre estos temas ha sido autor de varios libros y diversos artículos en revistas y congresos nacionales e internacionales.



**Jorge Eduardo Ibarra Esquer** es Maestro en Ciencias por el Centro de Investigación Científica y de Educación Superior de Ensenada. Se desempeñó como coordinador del Programa Educativo de Ingeniero en Computación de UABC. Actualmente,

es profesor titular de tiempo completo y desarrolla líneas de investigación en enseñanza de la programación, sistemas electrónicos digitales e internet de las cosas. Pertenece a la Red Temática Mexicana de Ingeniería de Software.



**Félix Fernando González-Navarro** es Doctor en Inteligencia Artificial por el Departamento de Lenguajes y Sistemas Informáticos de la Universitat Politècnica de Catalunya.

Actualmente, es investigador titular de tiempo completo en la UABC liderando el Laboratorio de Inteligencia y líder del Cuerpo Académico de Cómputo Científico ante PRODEP-SEP. Pertenece a la Red Temática Mexicana de Ingeniería de Software y es miembro del Sistema Nacional de Investigadores (SNI) nivel 1.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

## BIOMÉDICA

*Recibido 18 Oct 2016 Aceptado 24 Oct 2016*

*ReCIBE, Año 5 No. 3, Noviembre 2016*

# Potencial Eléctrico en el Corazón: Representación Mediante un Grafo

**Heart Electrical Potential: a Graph Representation**

Natalia González-Cervantes<sup>1</sup>

nataliaglezcervantes@gmail.com

Aurora Espinoza-Valdez<sup>1</sup>  
aurora.espinoza@cucei.udg.mx

Ricardo Salido-Ruiz<sup>1</sup>  
drsalidoruiz@gmail.com

<sup>1</sup> Depto. de Ciencias Computacionales, CUCEI, Universidad de Guadalajara, Guadalajara, Jalisco, México.

**Resumen:** De acuerdo a la Organización Mundial de la Salud (OMS), las enfermedades cardiovasculares constituyen un problema de salud pública por su alta frecuencia, además de que son la principal causa de muerte en todo el mundo. Motivo por el cual es necesario investigar nuevos enfoques de prevención y tratamiento de las complicaciones cardiovasculares. En este trabajo se estudia el potencial eléctrico en el corazón empleando elementos de la teoría de grafos, es decir representando las estructuras cardíacas involucradas y sus conexiones por medio de vértices y aristas. Los resultados muestran que es posible describir el potencial eléctrico en el corazón incorporando información fisiológica en las aristas del digrafo. Esto es, el digrafo puede proporcionar información con respecto al potencial eléctrico por regiones para una mejor localización.

**Palabras clave:** Grafo, corazón, potencial eléctrico.

**Abstract:** According to the World Health Organization (WHO), because of its high frequency, cardiovascular diseases are a public health problem, moreover they are the leading cause of death worldwide. Then, it is necessary to investigate new approaches to prevention and treatment of cardiovascular complications. In this paper we studied the electrical potential in the heart using elements of graph theory, i.e., representing cardiac structures involved and their connections through vertices and edges. The results show that it is possible to describe the electrical potential in the heart including physiological information

into the edges of the digraph. That is, the digraph can provide information regarding the electrical potential in certain regions for better localization.

**Keywords:** Graph, hearth, electrical potential.

## 1. Introducción

A través de los años, la comprensión del ser humano desde un punto de vista biológico (es decir el entendimiento del funcionamiento de los sistemas y aparatos que lo componen) se convirtió en un objetivo primordial y desafiante para la ciencia debido a la complejidad de los mismos. En la actualidad sabemos que una ciencia no es capaz de describir con detalle todos los aspectos relacionados a los sistemas biológicos. Ciencias como la biología y la computación se fusionan en lo que conocemos como biología computacional con el objetivo de estudiar a detalle todos los aspectos relacionados con éstos sistemas. De manera más concreta, la biología computacional tiene como objetivo el proponer modelos que ayuden a describir los procesos biológicos en términos del lenguaje computacional con el fin de pronosticar o predecir el comportamiento de éstos sistemas (Lobardero, 2014 y Shapiro, 1975). Por otro lado, el modelo describe teóricamente un objeto de la realidad mediante elementos dentro del campo de la computación. El éxito o fracaso de éste modelo dependen de la precisión con la que se construya ésta representación numérica. Otro aspecto importante que interviene es la fidelidad con la que se concreten hechos y situaciones naturales en forma de variables y la relación entre ellas. La utilidad del modelo radica en que ayuda a estudiar cómo se comportan las estructuras complejas frente a aquellas situaciones que no pueden verse con facilidad en el ámbito real. Desde mediados de los 90's una rama de la computación ha sido utilizada con este objeto: la teoría de grafos; la cual trata a los sistemas complejos de una manera más sencilla reduciéndolos a componentes (vértices) e interacciones entre ellos (aristas) (Lobardero, 2014).

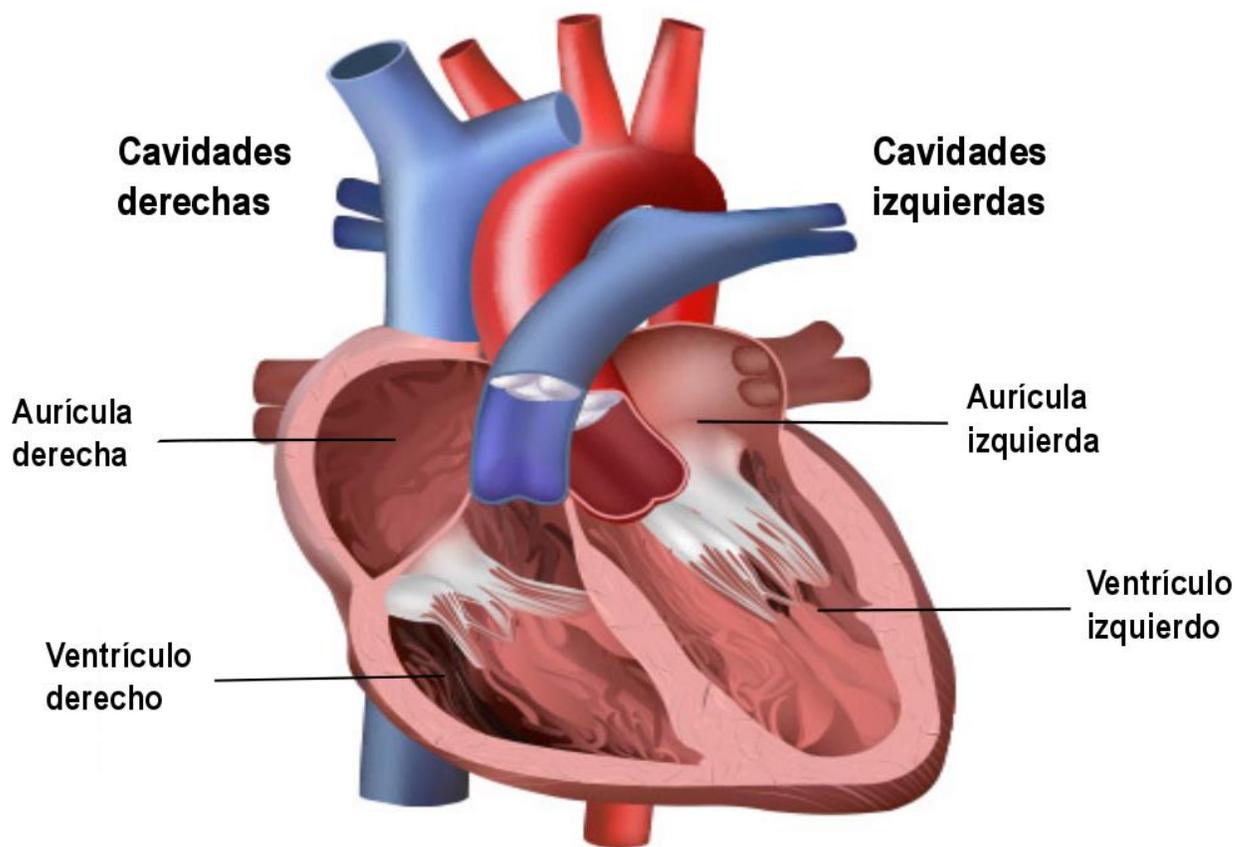
Por otro lado, el funcionamiento del corazón ha sido objeto de una serie de estudios (Peñaranda, 2012, Jiang, 2010, Min-Hung, 2016, Göktepe, 2009, Villa, 2000, Basavaprasad, 2014), esto porque el corazón es un órgano vital. Entre estos estudios realizados utilizando grafos, destaca el trabajo de Villa y Reyna (2000), donde se desarrolló un algoritmo utilizando grafos borrosos con el objetivo de simplificar los métodos tradicionales de caracterización de señales biológicas en especial la del trazo ECG (Villa, 2000). Más recientemente, Basavaprasad y Ravindra (2014) explican diferentes tipos de grafos y presentan un modelo de red teórica de gráfico en el corazón humano, donde se proponen diversos grafos que muestran las posibles rutas que podría tener el flujo sanguíneo en el corazón. Basavaprasad y Ravindra proponen solucionar problemas de la red cardiaca micro-densa modificando mediante los grafos de Euler y Hamilton la circulación del Sistema Cardiaco (Basavaprasad, 2014). Como se puede observar, sí existen estudios del corazón utilizando la teoría de grafos; sin embargo no hemos encontrado ningún estudio referente al potencial eléctrico del corazón utilizando grafos.

## 2. El Corazón

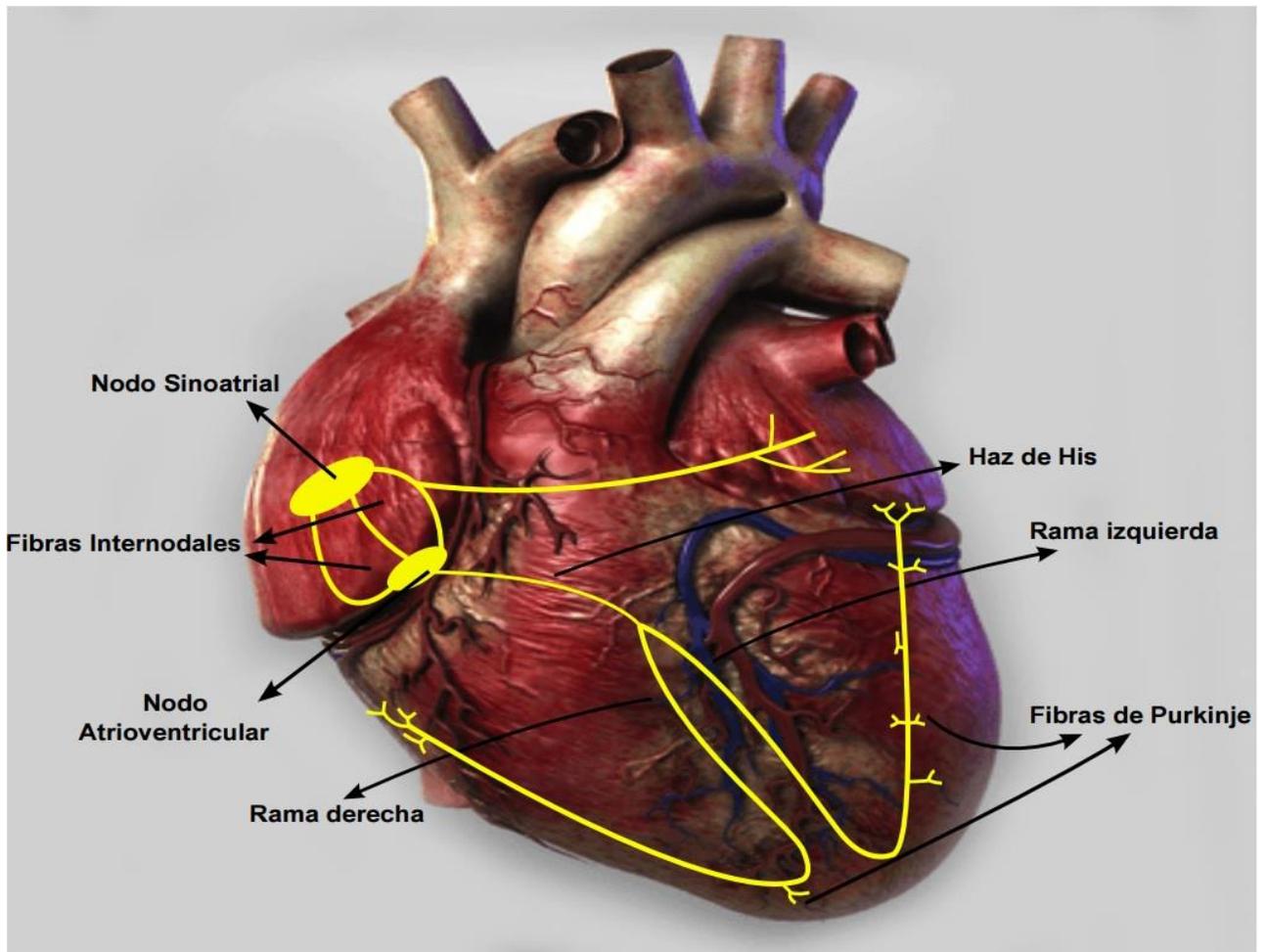
El órgano del corazón está formado por músculos cardíacos que generan impulsos eléctricos. Este órgano tiene la función de oxigenar cada célula de nuestro cuerpo. La complejidad del corazón va más allá de solo bombear sangre; requiere de procesos biofísicos que lo ayudan a ser un órgano autómatas latiendo de esta manera por sí solo y recibiendo ocasionalmente órdenes del sistema nervioso central y del sistema endócrino. Podríamos preguntarnos, ¿qué tiene que ver la computación con el corazón? Para poder comprender la relación que existe, es necesario comprender a detalle el funcionamiento del corazón.

La anatomía del corazón consta de 4 cavidades: 2 aurículas y 2 ventrículos, la sangre entra por las aurículas al corazón y de las aurículas la sangre es enviada a los ventrículos desde donde será expulsada la sangre para enviarla al resto del cuerpo (ver Fig. 1) (Uribe, 2014). Por otro lado, el músculo cardiaco

(miocardio) es el que hace latir al corazón, tiene la capacidad de originar y transmitir sus propios impulsos eléctricos para contraerse a través de células especializadas. Esta señal eléctrica se origina en el nodo sinoatrial, este se encuentra ubicado en la parte superior derecha del corazón (véase Fig. 2). Al nodo sinoatrial también se le denomina el marcapasos natural del corazón. Cuando el marcapasos emite el impulso eléctrico, éste estimula la contracción de las cavidades superiores del corazón (aurículas), después la señal pasa por el nodo atrioventricular (AV). El nodo AV detiene la señal un instante y la envía por las fibras musculares de las cavidades inferiores (ventrículos) estimulando su contracción (Uribe, 2014).



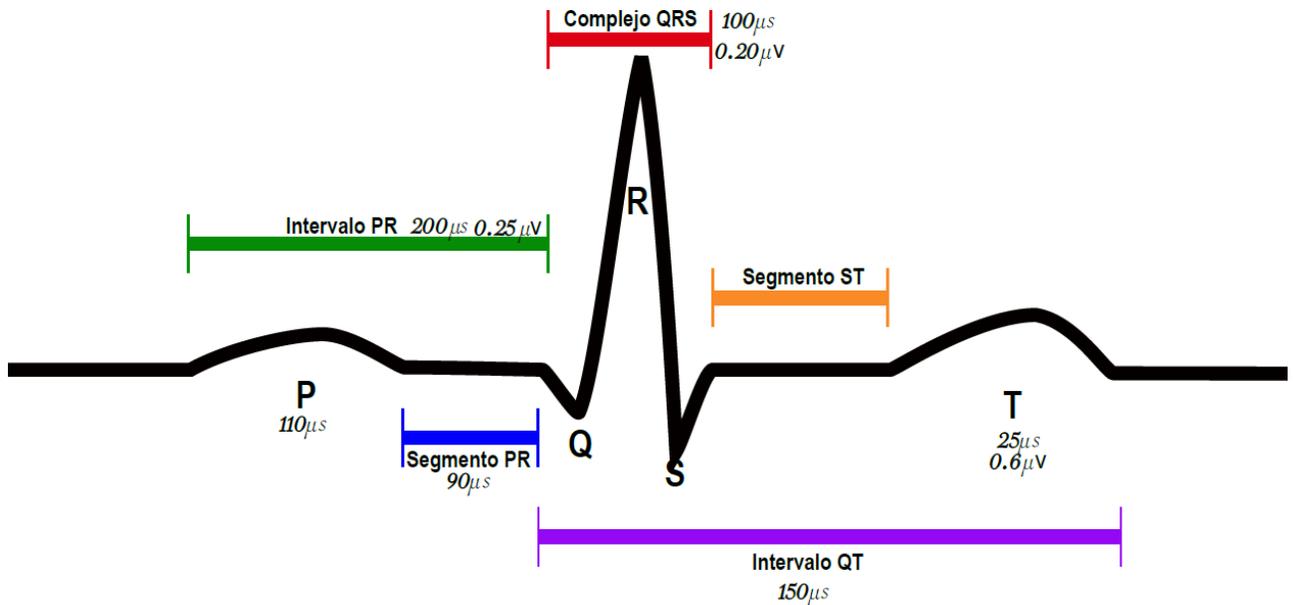
*Figura 1: Anatomía del corazón (imagen modificada de bioedonline.org).*



*Figura 2: Potencial eléctrico del corazón (imagen modificada de healthncure.org)*

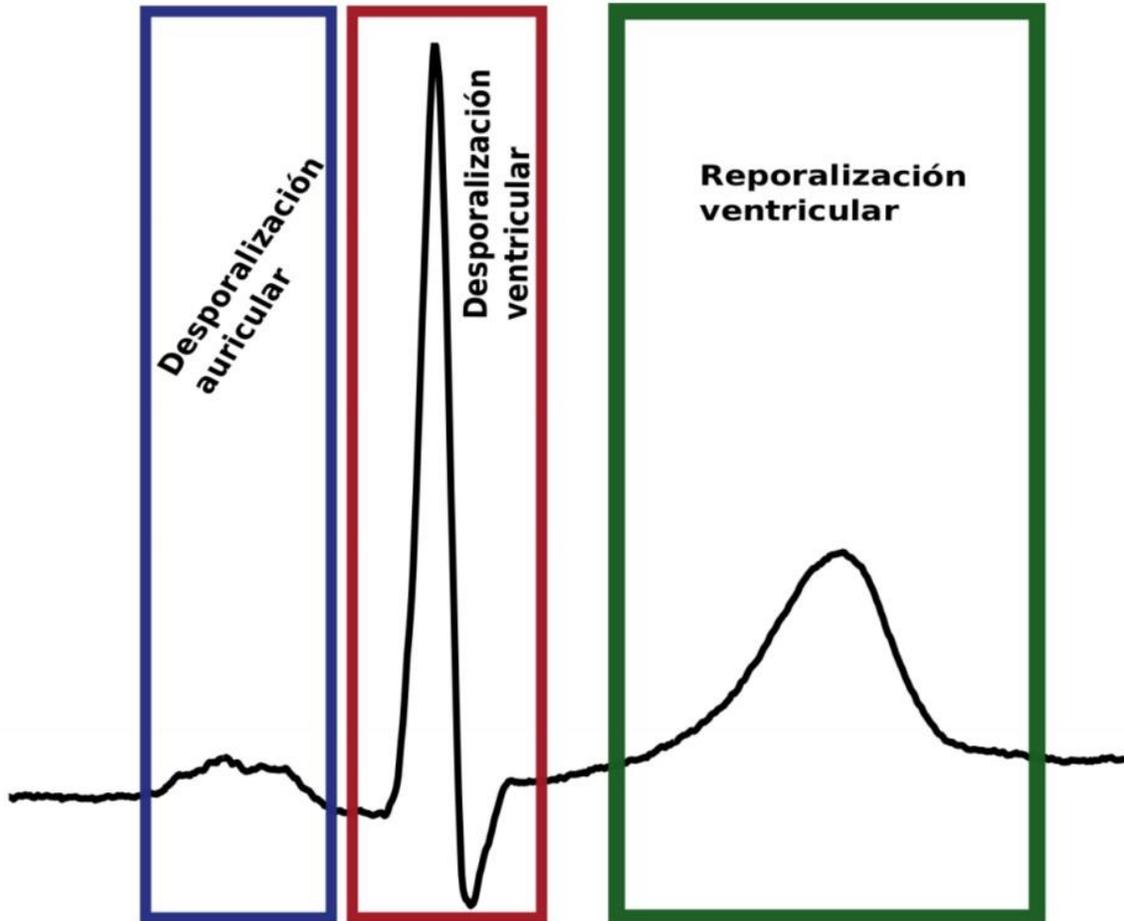
A través de un electrocardiograma (ECG) es posible ver la representación gráfica de los movimientos del corazón, ya que este estudio puede registrar la actividad eléctrica muscular que causa los movimientos del corazón [6]. El resultado del ECG se puede observar en la Fig. 4, donde la primera ondulación pequeña en la parte superior del trazado la cual se denomina onda P e indica que las aurículas son estimuladas en forma eléctrica para bombear la sangre hacia los ventrículos. El siguiente segmento se le conoce como complejo qRs y nos indica que el impulso eléctrico llegó a los ventrículos y generó la contracción de los mismos. La siguiente ondulación es la onda T y nos dice que el corazón se

recupera del impulso que recibió, es decir; se encuentra relajado y listo para el siguiente impulso eléctrico (Shapiro, 1975, Uribe, 2014).



**Figura 3:** Señal electrocardiográfica del corazón.

Los impulsos eléctricos se producen a través de la despolarización y repolarización de las células cardíacas, es decir, son células excitables que en respuesta a un estímulo generan un potencial de acción asociado a una respuesta contráctil. Un potencial de acción es un cambio reversible en el potencial de la membrana producida por la activación secuencial de diversas corrientes iónicas generadas por la difusión de iones a través de la membrana a favor de su gradiente electroquímico. Así, durante la despolarización la célula pasa de estar interiormente con carga negativa a positiva, en otras palabras, las células tienen en su interior una carga negativa y durante el potencial de acción entran iones positivos a la célula, esto es la despolarización y provoca las contracciones musculares cardíacas (sístole) y la repolarización es cuando la célula recupera su carga interior negativa, es cuando se relaja el músculo (diástole), ver Fig. 4.



*Figura 4: Esquema la señal cardiaca y su relación con las fases del corazón.*

### 3. Resultados

A fin de visualizar la topología de potencial eléctrico en el corazón podemos modelar el potencial eléctrico del corazón en el cual se puedan incluir leyes fisiológicas a partir de la definición de un grafo (Diestel, 2005). El potencial eléctrico se puede representar a través de un grafo donde: cada vértice representa los puntos principales que generan los impulsos eléctricos y que llevan la electricidad a cada parte del corazón, las aristas describen el valor máximo de voltaje y su duración en tiempo que descarga cada vértice.

**Definición 1:** Un grafo  $G$  es una terna ordenada  $(V, E, \Psi)$  que consiste en un conjunto no vacío  $V$  de vértices, de un conjunto  $E$  de aristas y de una función de incidencia  $\Psi: E \rightarrow K \leq 2$ , donde  $K \leq 2$  es subconjunto de  $V$  de tamaño menor o igual a 2, es decir, cada arista cumple una de las siguientes condiciones:

- $\Psi$  asocia la arista  $e$  a un par de vértices de  $V$ ,  $\Psi(e) = \{u, v\}$ .
- $\Psi$  asocia la arista  $e$  a un vértice de  $V$ ,  $\Psi(e) = \{u\}$ .

$G$  representa el grafo de potencial eléctrico en el corazón. Los vértices de  $G$  tienen todas sus aristas orientadas en tal forma que simboliza la dirección de la circulación del potencial eléctrico en el corazón. Además, cada arista es etiquetada con la información fisiológica por la función  $\Psi: E \rightarrow (R^+ \cup 0)^2$  donde es el 2 número de parámetros fisiológicos considerados para el desarrollo del potencial eléctrico del corazón.

Cada arista es etiquetada como  $e_i$  donde  $i \in \mathbb{N}$ , ahora cada arista es etiquetada como  $e_i(t_i)$  donde  $t_i$  es el tiempo. El tiempo del ECG se estimó el intervalo de tiempo que debe transcurrir desde un vértice a otro, se debe tomar en cuenta que diferentes factores influyen en el cálculo de la despolarización y repolarización auricular y ventricular, por lo que no es posible determinar con exactitud el tiempo o el voltaje de cada una de las ondas, pero se puede estimar el máximo y mínimo voltaje y tiempo que debe tener cada una, y mientras se respeten estas pautas, se puede decir que se tiene un ciclo normal.

Comenzando con el marcapasos natural del corazón (nódulo sinusal), el impulso eléctrico recorre las fibras internodales hasta llegar al nodo atrioventricular, en el grafo este recorrido es representado por los vértices  $v_1, v_2, v_3, v_4$  y  $v_5$  los cuales simulan la primera onda, la onda P, la cual tiene un tiempo máximo entre  $0.10'' - 0.12''$ . Una vez que la onda eléctrica viaja hasta el nodo atrioventricular se retiene la señal por al menos  $0.8''$  en el vértice  $v_6$ , donde  $e_{av}$  representa un tiempo de duración de  $.9''$ , esto con el fin de permitir a las aurículas y ventrículos terminar su proceso de llenado y vaciado, una vez completado esto, el impulso

sigue su camino hacia el tronco del Has de His en el vértice v7 y comienza de esta manera la despolarización ventricular, donde la corriente eléctrica se divide entre las dos ramas del Has de His (izquierda v8 y derecha v9), teniendo en cuenta que en la rama izquierda predomina el potencial eléctrico debido a dos ramas más que se encuentran en este lado llamadas: división posterior y división anterior (v10, v17, v18) el recorrido por estas ramas genera la onda R de nuestro electrocardiograma, siendo por este motivo una deflexión positiva larga, el viaje eléctrico continua hasta las fibras de Purkinje {v12, v15, v15, v20, v24, v23, v11, v13, v14, v19, v21, v22, v22, v25, v26, v27, v28,v29,v30} las cuales generan la onda s terminando de esta manera la despolarización ventricular y el complejo qRs que no debe medir más de 0.10". Las fibras de purkinje forman parte del sistema bioeléctrico del corazón y se distribuyen por todo el miocardio ventricular. La repolarización ventricular comienza al terminar el complejo qRs generando de esta manera la onda T de nuestro ECG y propicia el término de un ciclo eléctrico para prepararse para el siguiente.

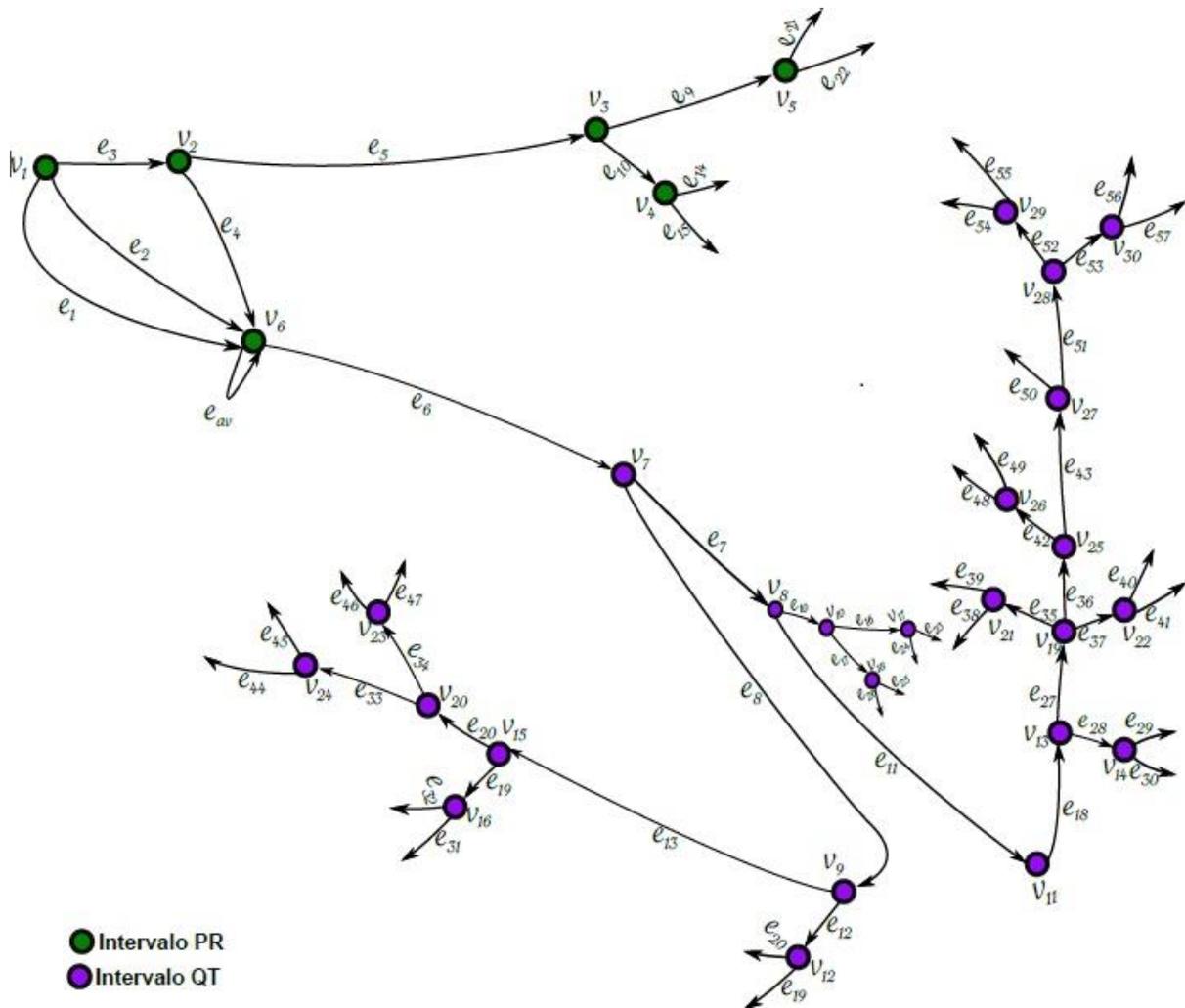
Una vez generado el grafo principal del corazón sano se pueden analizar diferentes aspectos en el comportamiento del corazón e incluso proponer soluciones a alguna de las problemáticas que presenta, como las arritmias.

## 4. Conclusiones

El corazón es un órgano complejo en cuestión de funcionalidad y estructura. En este trabajo se estudió el potencial eléctrico en el corazón utilizando teoría de grafos. El estudio muestra que se pueden incorporar las leyes fisiológicas involucradas. Esto es, cada uno de los vértices representa uno de los puntos principales que generan los impulsos eléctricos y que llevan la electricidad a cada parte del corazón, las aristas describen el valor máximo de voltaje y su duración en tiempo que descarga cada vértice. Además, puede proporcionar información con respecto al potencial eléctrico por zonas para una mejor localización.

Como trabajo a futuro, será incorporando todo a un sistema y programar el grafo para generar el ECG de un sujeto sano y el conjunto de combinaciones fisiológicas que describen la patología de arritmias.

**AGRADECIMIENTOS:** Un especial agradecimiento al Médico Internista. Dr. Vitelio Augusto Mariona Montero por asesoría en la fisiología del corazón.



**Figura 5:** Grafo del potencial eléctrico en el corazón. Donde el intervalo PR representa la despolarización auricular y el intervalo QT la despolarización y repolarización ventricular.

# Referencias

Basavaprasad B., Ravindra S. Hegadi. A graph theoretical network model on human heart. International Journal of Applied Engineering Research. Vol. 9 No. 20 pp 4549-4553, 2014.

Diestel R. Graph Theory. Springer-Verlag Heidelberg, New York, 2005.

Göktepe S., Kuhl E. "Computational modeling of cardiac electrophysiology: a novel finite element approach" Int. J. Numer. Methods Eng. 79, pp. 156-178, 2009.

Jiang Z., Connolly A. and Mangharam R. "Using the virtual heart model to validate the mode-switch pacemaker operation", Proc. IEEE Eng. Med. Biol. Soc. Conf., pp. 6690-6693, 2010.

Lombardero O. A. Un vistazo a la Biomatemática. Números: Revista de Didáctica de las Matemáticas. Vol. 86, pp 29-38, 2014.

Min-Hung C., Po-Yuan C. and Ching-Hsing L., "Quadratic adaptive algorithm for solving cardiac action potential models". Computers in Biology and Medicine, vol. 7, pp. 261-273, 2016.

Peñaranda A., Cantalapiedra I, Bragard J. and Echebarria B. "Cardiac dynamics: A simplified model for action potential propagation", Theo Bioi and Med Modelling, vol. 9, pp. 50, 2012.

Shapiro M., Martínez S. J. Arritmias cardiacas: Introducción a su Diagnóstico y Tratamiento. Librería de Medicina, pp 27-31, 1975.

Uribe W. Electrocardiografía Clínica. De Lo Básico A Lo Complejo. Bogotá, 2014.

Villa Angulo C., Reyna Carranza, M.A. Algoritmo para detectar el complejo QRS en el trazo de ECG, utilizando operaciones para grafos borrosos. Memoria Electro-Congreso Internacional de Ingeniería Electrónica. Vol. 22 pp 109-114, 2000.

<http://www.who.int/en/>

## Notas biográficas:

**Natalia González-Cervantes** es estudiante del quinto semestre de la carrera de Ingeniería en Computación en la Universidad de Guadalajara. Edad: 24 años. Intereses: Seguridad web, redes computacionales, biología y matemáticas.

**Aurora Espinoza-Valdez** recibió el grado de Licenciada en Matemáticas Aplicadas en la Universidad Autónoma de Baja California (UABC), México. El grado de Maestra en Ciencias Aplicadas en la Universidad Autónoma de San Luis Potosí (UASLP), México. El grado de Doctora en Ciencias en el Instituto Potosino de Investigación Científica y Tecnológica (IPICYT), México. Desde el 02/2012 es Profesor Investigador del Depto. de Ciencias Computacionales en el Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara. Su interés de investigación es el modelamiento matemático de sistemas biológicos y sistemas dinámicos.

**Ricardo Salido-Ruiz** es Ingeniero en electrónica por la Universidad Autónoma de Baja California. Trabajó en el sector privado en Electrónica Lawrence en el diseño y creación de prototipos de circuitos con componentes de montaje superficial. Realizó su maestría en Bioelectrónica en CINVESTAV-IPN en el área de procesamiento de señales de transductores electroacústicos. Realizó sus estudios de doctorado en Instituto Nacional Politécnico de Lorraine, INPL (hoy Universidad de Lorraine) en Francia para en el área de procesamiento de señales electrofisiológicas. Ahí trabajó con señales electroencefalográficas EEG, provenientes de pacientes con epilepsia. Laboró en el programa de Bioingeniería de la Universidad Autónoma de Baja California como Profesor Titular A. Presidió el comité científico evaluador de trabajos libres y del concurso estudiantil de la SOMIB en el congreso nacional de ingeniería biomédica CNIB2014 en Puerto Vallarta. Actualmente se desempeña como profesor de tiempo completo en el Departamento de Ciencias Computacionales del CUCEI.



Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 2.5 México.